DOCUMENT RESUME

ED 140 860 IR 004 997

AUTHOR Stirling, Keith H.
TITLE The Effect of Document Ranking on Retrieval System Performance: A Search for an Optimal Ranking Rule.
PUB DATE May 77
NOTE 145p.; Ph.D. Dissertation, University of California

EDRS PRICE MF-$0.83 HC-$7.35 Plus Postage.
DESCRIPTORS Algorithms; *Information Retrieval; Information Seeking; Input Output; *On Line Systems; *Performance; Relevance (Information Retrieval); *Search Strategies; Use Studies

ABSTRACT

As the use of online literature searching systems increases, the need for reducing the amount of nonrelevant material encountered by the users becomes more apparent. One method for achieving greater selectivity in document retrieval is by using relevance (or utility) estimates from past users to estimate document values for future users. However, ranking rules used to sequence the retrieved documents have not provided optimal performance. This research attempts to develop an optimal document ranking rule and contrast it with other document ranking algorithms. Three ranking algorithms are studied: (1) the probability ranking rule (PRR), (2) the optimal probabilistic strategy (OPS), and (3) the optimal ranking rule (ORR). The study also identifies several computationally difficult problems associated with information retrieval. (Author/DAG)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED140860

THE EFFECT OF DOCUMENT RANKING ON RETRIEVAL SYSTEM PERFORMANCE: A SEARCH FOR AN OPTIMAL RANKING RULE

PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY

Keith H. Stirling

TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER.

Keith H. Stirling

School of Library and Information Studies
University of California
Berkeley, California

May 1977

IR004991

The Effect of Document Ranking on Retrieval System Performance: A Search for an Optimal Ranking Rule

Copyright © 1977

by

Keith H. Stirling

The Effect of Document Ranking on Retrieval System Performance:
A Search for an Optimal Ranking Rule

By

Keith Henry Stirling

B.S. (Brigham Young University) 1964
M.S. (Brigham Young University) 1966

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Librarianship

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

William S. Cooper .....

Richard M. Karp .....

M. S. Maron .....

Committee in Charge

.....

The Effect of Document Ranking on Retrieval System Performance: A Search for an Optimal Ranking Rule

Abstract

Keith H. Stirling

As the use of online literature searching systems increases, the need for reducing the amount of nonrelevant material encountered by the users becomes more apparent. One method for achieving greater selectivity in document retrieval is by using relevance (or utility) estimates from past users to estimate document values for future users. This sort of feedback has previously been used, notably, in test environments such as the Cranfield and SMART retrieval systems. Using the feedback, numerous strategies were applied to these systems in an attempt to maximize the retrieval of relevant material. However, ranking rules used to sequence the retrieved documents have not provided optimal performance.

This research attempts to develop an optimal document ranking rule--assuming the availability of relevance feedback from system users--and contrast it with other document ranking algorithms. It will not only lay the basis for an optimal rule, but will also supply a standard against which the effectiveness of other ranking procedures may be established.

Three ranking algorithms are studied. They are the probability ranking rule (PRR), the optimal probabilistic strategy (OPS), and the optimal ranking rule (ORR).

The PRR arranges items in order of their decreasing probability of yielding the needed information. It is commonly believed that this

represents the optimal ranking procedure. However, this research demonstrates that the PRR will guarantee an optimal document ranking only when (1) no more than one relevant document is required to satisfy the information needs of any one patron, or (2) no more than a total of two documents are retrieved. The reason the PRR produces less than the best ranking is that it is indifferent as to whether or not a given document in a sequence will be terminal or nonterminal for a given user. Determining a document's optimal position within a sequence cannot be done solely on the basis of the number of users who regard it to be relevant. Data demonstrating the degradation of the PRR's performance as a ranking rule were generated via simulation. Multiple regression analysis of these data indicated that if the ratio of system "user-types" to numbers of retrieved documents is at least three to one, the PRR's performance will be degraded by no more than 10 percent. The implication is that retrieval systems, which accumulate relevance feedback estimates from large numbers of users (consistent with the above ratio), are capable of providing near optimal document rankings for future users when using the PRR.

The OPS yields a document ranking that minimizes the users' expected loss. Although resulting in an optimal document ranking, the algorithm is shown to require $MN! \approx M(N)^N$ cost comparisons for solution. (M equals the number of user-types, N equals the number of documents.) For even moderate values of N, the number of computations required exceed the capacities of sophisticated computers.

In contrast, the ORR, which uses a backtracking solution technique, requires $MN[2^{N-1}-1]$ cost comparisons to achieve the optimal document ranking. Altnough the number of cost comparisons associated with the ORR

represents a lower order exponential, it, nonetheless, is exponential and is practical only for smaller numbers of retrieved documents.

The cost/benefit breakeven point was determined between the PRR and the ORR. Let $P = cI - \alpha W$ where I equals the average difference in the number of nonrelevant documents avoided per user when comparing the ORR and the PRR, and W equals the difference in the number of cost computations required by the two procedures. "c" and "α" represent constants that transform "nonrelevant documents avoided" and "number of cost computations" into units of utility to the user. The negation sign preceding αW indicates that this factor contributes a loss to the user's utility function. The breakeven point varies for both numbers of documents and users, but analysis indicates that for the ORR to be cost effective, the value of c/α must, at least, be greater than 10^3.

In summary, this research has identified the limitations of the probability ranking rule. It has also provided an optimal document ranking procedure as a standard for comparison. Moreover, the study identifies several computationally difficult problems that are associated with information retrieval. If a more cost effective, improved ranking rule is to be realized, the consideration of heuristic solutions is recommended. Although they may not guarantee optimal document ranking, they will very likely provide a better cost/benefit ratio compared with the PRR and not involve the expense associated with computing the ORR.

William S. Cooper

William S. Cooper, Chairman

ACKNOWLEDGEMENTS

The author appreciates greatly the guidance and encouragement given him by his committee chairman, William S. Cooper. Numerous discussions with him and his own research and writing have provided the catalyst for this research.

Receiving his first real exposure to Bayesian probability theory from M. E. Maron, the author also gratefully acknowledges the beneficial discussions with him. In addition the author expresses thanks to Richard M. Karp for serving on his dissertation committee.

Also contributing to the completion of this dissertation have been the faculty of the School of Library and Information Sciences at Brigham Young University. Their occasional prodding, willingness to adjust teaching loads, and review chapters has not been casual. Special thanks are expressed to Maurice P. Marchant for his helpful suggestions during the preparation of the manuscript and to Donald Kraft for his comments.

Adding secretarial support to the faculty have been the office staff of the School. Particular appreciation go to Vilia D. Van Ausdal and Jean Butler for their patience and meticulous effort in typing this manuscript. The superscripts and subscripts within equations are an anathema to any typist.

Without the financial resources of both the School and Computer Services at Brigham Young University, supplied in the form of released research time and computer time, the dissertation would not have been completed. In addition, this author is deeply appreciative of the Higher Education Act, title II-B Fellowship support received while a doctoral student at the University of California. These financial

commitments to the author's training and research have been substantial.

None of my children can remember a time when I was not either going to school or working on my dissertation. I do appreciate their willingness to "share" their father's time.

Words are inadequate to acknowledge the total support received from my wife, Dolores. Without her encouragement this research would neither have been completed nor started.

Finally, I would be ungrateful not to acknowledge the encouragement I have received from my mother, Mrs. Delbert Stirling. She has bolstered my faith by the manner in which she has met the challenges occurring in her own life.

TABLE OF CONTENTS

Chapter		Page
	ACKNOWLEDGEMENT	i
	LIST OF FIGURES	v
	LIST OF TABLES	vii
	LIST OF APPENDIXES	viii
1.	INTRODUCTION	1
	Statement of the Problem	
	Illustrating the Problem by an Example	
	The Notion of Search Strategy	
	Ranking Retrieved Output	
2.	A SEARCH FOR AN OPTIMAL RANKING RULE	17
	The Retrieval Rule	
	The Stop-Search Point	
	A Measure of Retrieval Effectiveness	
	Testing the Ranking Rules	
	A Counterexample to the Optimality of the PRR	
	Objection #1 (different stop-search values)	
	Objection #2 (ranking and optimizing criteria different)	
	Objection #3 (an optimal probabilistic strategy)	
	Objection #4 (value of retrospective retrieval)	
	Objection #5 (majority of users penalized)	
	Objection #6 (on having a priori stop-search points)	
3.	THE LIMITS OF PROBABILITY RANKING	36
	Case I: Stop-Search Point Equivalent to One Relevant Document	
	Case II: Unrestricted Stop-Search Point	
	Case III: Only Two Documents Are Retrieved	
4.	SIMULATING PROBABILITY AND OPTIMAL RANKING	49
	Simulating the Problem	
	Results of the Simulation Experiment	
	Validating the Simulation	

5. THE OPTIMAL RANKING ALGORITHM 61

Formulating the Problem
Staging the Solution
An Example Using the Staging Solution
The Optimal Algorithm
Analysis of the Optimal Algorithm
NP-Complete Problems

6. THE OPTIMAL RANKING RULE--REVISITED 79

Cost/Benefit Analysis
Objection #7 (estimation of ordinal statistics)
Objection #8 (relevance versus utility estimates)

7. CONCLUSION 89

Suboptimal Performance of the Probability Ranking Rule
Computational Constraints on Optimal Ranking Procedures
A Compromise
In Retrospect

APPENDIXES 99

BIBLIOGRAPHY 122

LIST OF FIGURES

Figure		Page
1-1	Simple Document Retrieval System	3
1-2	A Document-User Matrix Representing the Different Information Needs of Nine Types of Users Having the Same Request	6
2-1	A Document Ranking According to the PRR	22
2-2	Document Ranking Produced by the ORR	24
2-3	Document Ranking According to the PRR	25
2-4	A Counter Example to the PRR where Each User Has Equal Stop-Search Points	25
3-1	Example #1: An Application of Equation (7) (Cardinally Decreasing Sequence)	44
3-2	Example #1: An Application of Equation (7) (Inverted Sequence)	44
3-3	Example #2: An Application of Equation (7) (Cardinally Decreasing Sequence)	45
3-4	Example #2: An Application of Equation (7) (Inverted Sequence)	45
3-5	Example #3: An Application of Equation (7) (Cardinally Decreasing Sequence)	46
3-6	Example #3: An Application of Equation (7) (Inverted Sequence)	46
4-1	Probability and Optimal Ranking Search Length Agreement Versus Numbers of Users and Documents	52
4-2	Simulated Data Comparing Optimal and Probability Ranking (Numerical Improvement)	54
4-3	Simulated Data Comparing Optimal and Probability Ranking (Percent Improvement)	56
5-1	An Optimal Four-Document Sequence By The Backtracking Method	65

Figure		Page
5-2	Document Ranking Produced by the PRR	67
5-3	Document Ranking Produced by the ORR	70
5-4	Optimal Ranking Algorithm Time and Memory Requirements Versus Number of Documents	75
6-1	Number of Documents (N), Number of Users (M), and Coefficient Ratio (C/α) at which the Performance Function (P) Breaks Even when Comparing the Optimal Ranking Rule and the Probability Ranking Rule .	81
E-1	All Possible Configurations of Relevant Documents in a Two-Document Sequence (Decreasing Cardinality) . .	116
E-2	All Possible Configurations of Relevant Documents in a Two-Document Sequence (Inverted Sequence)	116

LIST OF TABLES

Table		Page
5-1	Second Stage Decision Table	67
5-2	Third Stage Decision Table	68
5-3	Fourth Stage Decision Table	69
5-4	Fifth Stage Decision Table	69

LIST OF APPENDIXES

APPENDIX A: ALGOL PROGRAM FOR DETERMINING THE ORR 99

APPENDIX B: ALGOL PROGRAM FOR SIMULATING DOCUMENT RETRIEVAL 104

APPENDIX C: PROGRAM FOR DETERMINING MAGNITUDE OF STORAGE REQUIREMENTS DURING AND AT END OF EACH STAGE OF THE ORR 108

APPENDIX D: APPLICATION OF EQUATION (6), CHAPTER THREE 109

APPENDIX E: DEVELOPING EQUATION (7), CHAPTER TWO UNDER THE CASE II ASSUMPTION 111

APPENDIX F: DEVELOPING EQUATION (6), CHAPTER FIVE 119

Chapter 1

INTRODUCTION

Most of us have made decisions which, had we had access to additional data, would have been altered. And yet, we are frequently faced with too much information or our retrieval system provides us with information of the wrong kind. In fact, encountering erroneous or misinformation is sometimes more detrimental than receiving no information. Decrying the information rich culture in which we live, Taylor states that the mark of a good information retrieval system should be measured not by how much material is retrieved but rather by the material which is withheld from the user's view. He argues that better filters are needed to condense and compact retrieval system output. [1, pp. 1-5].

Bar-Hillel, on the other hand, resisted the general concern over exponential proliferation of publications by arguing that this avalanche condition would eventually result in a balance which would permit only the survival-of-the-fittest. [2]. In the past decade, however, direct access storage devices with capacities approaching one billion bytes per spindle cause one to wonder if the natural selection proposed by Bar-Hillel is being permanently postponed.

Scores of machine readable data bases boasting millions of bibliographic citations are now commercially available for on-line searching through time share networks. [3], [4]. The proliferation of communication carriers has accentuated the availability of these information-for-a-price

systems. [5]. In spite of technological advancement, however, development of information filtering techniques have not kept stride.

Even though all people should have realistic and convenient access to information resources for dealing with problems impacting in their lives, any mechanism for information transfer which simply yields an increased flood of citations--without a concommitant improvement in their selectivity--is likely to confuse rather than benefit its recipients. One way of achieving selectivity in document retrieval is to use document relevance estimates from past users to decide the likely value of the same material when retrieved for future users. Such retrospective retrieval systems have usually existed in test environments, such as the Cranfield and SMART collections, where numerous techniques have been devised for arranging retrieved documents in the "best possible" sequences. Although the ranking rules used to sequence the documents have rendered improved performance, they have not been "optimal".

Statement of the Problem

This research will attempt to develop an optimal document ranking rule and contrast it with other ranking algorithms that have been designed to rank documents by users' relevance judgments. Not only will the basis for an optimal ranking algorithm be laid, but a standard will be supplied against which the effectiveness of other document ranking rules may be established.

This chapter will describe a specialized retrieval system which will permit the illustration of a document ranking mechanism. In addition, important literature pertaining to search strategies and document ranking rules will be discussed. Finally, the "strategy" for the remainder of the report will be described.

Illustrating the Problem by an Example

A Selective Dissemination of Information (SDI) retrieval system is an example of a retrieval system which could utilize a document ranking rule. These SDI systems comprise scores of subject-specific, machine readable bibliographic records of journal articles from research literature. Appropriate subject descriptors have been assigned to each record and frequently an abstract exists for the entry. At a computer terminal, for example, a patron may retrieve the records by keying a combination of descriptors or keywords appropriate to his information need. The computer matches the terms contained in his request with the terms in the data base records and alerts the patron of any "hits". Figure (1-1) illustrates the retrieval system.

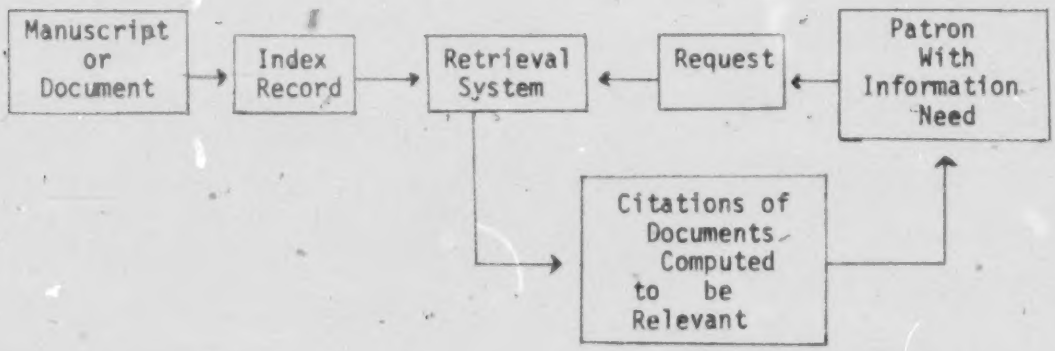


Figure 1-1 SIMPLE DOCUMENT RETRIEVAL SYSTEM

A logical fallacy, however, exists within this retrieval model. Consider the following inference:

Premises: { If a document, D, is to satisfy a patron's information need, then he must request it by submitting the request, R.
The patron submits the request, R.

Conclusion: The document, D, satisfies the patron's information need.

In sentential calculus this takes the form:

$$\frac{D \rightarrow R \quad R}{D}$$

Since the combined premises can be true while the conclusion is false, the inference is invalid. This means that while the SDI system may frequently supply useful information to the patron, without more sophisticated procedures he presently has little control of the retrieval product beyond "bibliographic shopping".

To improve the quality of information brought to the user's view, one might concentrate on the preparation of either the patron's requests or the document index records. One classic investigation into improving request representations is Taylor's sensitive elicitation procedure. [6]. Another classical method for creating document index records is Maron's problem-centering approach. [7]. Sometimes referred to as a literature searching system, this retrieval model has been extensively discussed by Salton [8] and Kochen [9, pp. 66-76].

Another approach for improving the quality of information supplied by the SDI system is to use feedback from past users as a filtering device. Doing this would screen some nonrelevant material from that supplied to future patrons who have similar information needs. Patron satisfaction with respect to retrieved information is best measured by the patron, himself. In the absence of his measurement, the best estimate is by other patrons who have had similar information needs and who have evaluated the retrieved information. Logically, a problem may still exist by inferring that one man's "reading" can be represented by another's but statistically the best estimate of the value one man places on a document can best be predicted from the collective values of those who

have similar information needs.

The data upon which a document ranking rule would be based can be collected in one of several ways. Upon reentering the SDI system at a later date, the system could invite the patron to evaluate each of the citations previously received with an appropriate response such as "relevant", "nonrelevant", or "not read". Another method would be for the user to receive by mail an evaluation form requesting the above information. Still another method for obtaining patron feedback would be to employ an investigator who would monitor the actions of the patrons throughout their entire experience with the retrieval system, including a followup evaluation as they examine the documents suggested by the system.[39, p. 89]. In all three approaches the patron would be informed that his evaluations, combined with those of other system users, will result in an overall improvement in the systems' retrieval effectiveness.

Now suppose that relevance data have been collected for the various requests received by the SDI system and that these data accurately reflect both the frequency and diversity of information needs of the user population. Suppose, further, that the matrix of data in Figure 1-2 represents responses to five retrieved documents by that part of the user population having the same request. Although the information needs of these users differed, in general many had needs that were the same or very similar. Grouping together those patrons who have the same needs permits one to reduce the entire population, in this example, into nine "user-types." An "x" represents a document relevant to the corresponding user-type.

The foregoing example has been developed to illustrate the benefit of providing documents to the users in a predetermined sequence. The

matrix of Figure 1-2 already proposes a logical order for searching the documents--namely, in a downward order beginning with document number one. This procedure follows a simple probability (or frequency) ranking rule which arranges items in order of their decreasing probability of yielding the needed information. If we could imagine that each user would complete his search after retrieving two relevant documents, then the sequence would require the patrons, collectively, to consider eighteen nonrelevant documents, or an average of two per patron. Each user-type is assumed to have equal weight. As a comparison, if the documents were considered in reverse order--starting with document number five--then an additional five nonrelevant documents would be encountered by the patrons, or an average of 2.56 per patron. Thus, the probability ranking rule provides a 20 percent average decrease in nonrelevant material. Certainly, a retrieval system which screens 20 percent of the nonrelevant documents encountered by the patrons, would be desirable.

User-types

Documents	1	2	3	4	5	6	7	8	9
1	x	x	x	x	x				
2						x	x	x	x
3	x			x			x		
4		x			x			x	
5			x			x			x

Figure 1-2 A DOCUMENT-USER MATRIX REPRESENTING THE DIFFERENT INFORMATION NEEDS OF NINE TYPES OF USERS HAVING THE SAME REQUEST

This example has attempted to illustrate that the order in which retrieved documents are arranged and subsequently passed to the patrons can substantially reduce the amount of nonrelevant material they must consider. The probability ranking rule is commonly believed to provide the optimal ranking procedure. It will be shown later, however, that a permutation of the sequence illustrated in Figure 1-2 will reduce the number of nonrelevant documents encountered by an additional 17 percent. The limitations of the simple probability ranking rule will be broached in chapters two, three, and four. Since the concept of document ranking is associated with the broader area of search strategy, it will be instructive to identify some of the pertinent literature.

The Notion of Search Strategy

In ranking the retrieved documents a sequencing strategy is developed in an attempt to minimize the number of documents which must be searched by each user type to satisfy its information need. The notion of sequential decision making in general overlaps with a number of subject disciplines and thus has a broader base for application than simply ranking documents. Even the example previously developed provides a different basis for applying this concept. For instance, once a librarian has obtained the requisite information from a patron, her skills are drawn upon to formulate a search strategy. The person's expertise may be used to form and modify a patron's request into a series of search expressions to be used in searching a data base composed of machine readable bibliographic records. Each of the expressions would be successively refined in an attempt to improve both the quality and quantity of retrieved information. Or on the other hand, using a knowledge of reference sources the librarian may need to devise a strategy for selectively

scanning them. The assumption seems reasonable that the efficiency of the strategy will be affected by the sequential procedure followed while improving the search expression and by the order in which the reference sources are consulted. The objective in these instances, then, would be to identify the optimal procedure for negotiating these searches.

Other meanings included under the mantle of search strategy that are suggested by Salton, and which lend themselves to optimization procedures within the automatic document processing environment are

> "... file generation systems which underlie all other processes; the construction of information products such as dictionaries and indexes that prove useful in many document processing environments; certain language processing routines which normally form the basis for automatic text processing and indexing systems; the query processing systems used to identify relevant information following a search of the stored records; the output display processing systems which interact with the system during information search and retrieval; and system evaluation and file maintenance facilities ..." [10].

Rocchio employs several of Salton's techniques to aid the user in formulating efficient search strategies. For example, an automated dictionary is used to cue the user regarding (i) synonyms relating to his search terms; (ii) the hierarchical nature of concepts; (iii) statistical co-occurrence between document and request terms. In addition, the request is "optimized" through "relevance feedback". Here the initial search request is automatically adjusted by promoting terms or concepts present in relevant documents and demoting those in nonrelevant ones. [11].

Some doubt has been cast, however, upon the effectiveness of these procedures. Chang (et.al) notes that the relevance feedback principle tends simply to move relevant documents already seen to the top of the stack rather than identify new relevant material. What the user really needs, he feels, is an improved request which will identify new relevant documents--not simply rearrange old ones. [12]. This problem would likely be less acute, however, if either the experimental data base included current awareness rather than strictly retrospective material, or different users with similar requests were using the retrieval system rather than having the same ones repetitively use the system.

Other areas where the optimization of search strategies is sought are in the formation of logical, boolean search expressions [13], [14], [15], and the binary search of data lists. [16]. (Although implied by Salton, the latter was not explicitly mentioned.) In addition, the literature on search strategy includes a number of ranking procedures that are of direct interest to the research reported here. Literature particular to document ranking algorithms will now be reviewed.

Ranking Retrieved Output

Perhaps the most common ranking criterion is that based upon a frequency interpretation to probability theory. Because of the general belief in its optimality as a ranking principle, the probability ranking rule will be the object of much of the analytical and empirical investigation reported in this research. To apply the rule a summary statistic for each document-request pair is computed by combining the numerical weights corresponding to the matching index and request terms. "The requestor would then be supplied with a list of references ranked according to the probability of their usefulness." [17]. Over the past decade there have

been numerous adaptations of this model. See, for example, [18], [19], [20], [21], [22], and [23].

Closely associated with the probability ranking criterion is the principal of sequential optimization in search theory. The latter term is usually associated with military search and reconnaissance operations, which are concerned with efficient search and detect patterns. [24]. Dobbie observes that two competing objectives are present in the target detection problem. One is to minimize the user's effort while locating the "target(s)" and the other is to maximize the probability of their detection. He further observes:

> "the expected effort is minimized by always distributing the effort to maximize the probability of detection with the effort expended thus far. This is a distribution requirement that is very difficult to meet. By contrast, the distribution that maximizes the probability of detection with a given effort can be non-optimal for all values of the effort less than the total effort; the effort can be applied by any schedule that finally attains the required distribution when all the effort has been applied." [25]. (emphasis added)

If the document ranking problem and sequential search theory have anything in common, the implication is that the solution is nontrivial and it may require considerable effort. However, reducing the search problem to a single target (i.e., a single document in our case), Leimkuhler shows that both the probability of success is always maximum and search effort is minimized when the items are "searched in decreasing order of their probabilities of yielding the information sought." [26]. (This result will be of interest when the limits of probability ranking are

explored in chapter three.) Such a ranking criterion requires that sufficient prior information exists to be able to rank all of the items in the file.

Another ordering criterion, with application to document retrieval, uses the principles of Bayesian statistical decision theory. A clear description of the technique is given in [27]. It can be shown that for any two events, A and B

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)} \qquad (1)$$

The conditional probability on the left, $P(A|B)$, represents the (posterior) probability of A given that B has occurred. $P(A)$ on the right represents the (prior) probability of A occurring. The advantage of Bayesian inference is that where prior information is available about the event A (for example, document relevance judgments), and related probability distributions are known for $P(B|A)$ and $P(B)$ (for example, the assignment of index terms to the documents), for each user's request an improved (posterior) estimate may be used to rank the retrieved documents. The disadvantage of using this approach is that as the number of events increases, the Bayesian expression becomes more complex--requiring knowledge about joint probability distributions. When applying Bayesian methods, the usual procedure is to (i) assume that the retrieval environment may be approximated by a known statistical distribution--thus permitting estimates of the prior probabilities (in a retrospective retrieval environment, the prior probabilities would be known), and (ii) assume some form of independence among the events--such as index terms occurring independently in documents.

Invoking both of the above assumptions Maron and Kuhns use the

indexer's estimate of the relevancy of term assignment to a document as an approximation of the conditional probability on the right of equation (1). Under their formulation the posterior probability is adapted to multiple term requests and thus can be used to rank the retrieved output for each user query. [28]. See also [29].

Tague maximizes the expected value of a utility function under her Bayesian approach to achieve optimal restructuring of the retrieved output. Her decision rule is based upon having a multivariate beta distribution for assumption (i) above, and a joint multinomial distribution for the conditional probabilities on the right of equation (1). She further avoids the necessity of determining joint distribution data by assuming (implicitly) the independence criterion and imposes a logistic model to transform the posterior Bayesian probability into a linear function. The posterior expected log odds for each relevancy rank for a document follow from this complex model. [30].

Deserving serious attention is the optimal document ranking rules, based upon Bayesian statistics, derived by Robertson and Sparck Jones. Starting with the "intuitively obvious ... probability ordering principle" they invoke the independence assumption (ii)--that index terms are independent--and apply the log-odds transformation to achieve a linear function of the Bayesian posterior probability. Either the latter function or a modification that includes a utility loss function may be used to rank the documents in order of their probability of relevance. The authors also clearly differentiate between retrospective retrieval with a test collection, and predictive retrieval in an interactive environment. In addition, they have apparently observed some anomalies to the "intuitively obvious" probability ranking principle since they add that the "probabil-

ities (must be) calculated from all available information." [31]. (Underscore is theirs.) However, since they summarize all individual user's satisfaction and dis-satisfaction into a single relevance judgment, their formulation would not apply in those cases where the differences between individual users' relevance judgments (for each document) are important.

Paralleling somewhat the results in chapters three and five of this report is Baker's reward-cost theory of behavior approach to an optimal ranking rule. Extending some of his previous work [32] and the work of Leimkuhler, he has used the fundamental design principle of sequential search theory which proposes that "expected search effort is minimized by always distributing the effort sequentially so as to maximize the probability of target detection with effort expended thus far." Baker notes a "distressing" example of suboptimal performance--yet the documents had been ranked in decreasing probability of user relevance. [33]. Although in general agreement with the results in chapter three, there is some variance regarding the efficiency of the optimizing algorithm in chapter five.*

Another type of document ranking rule is the "utility-theoretic" approach to document indexing. Here, retrieval effectiveness is optimized by assigning those index terms to the document, consistent with its having maximum utility after being retrieved. [34], [35]. Although providing an optimal ranking, the extensive calculations associated with this rule identify it as a member of a class of computationally difficult problems--

---

*Compare, for example, "An example using the staging solution" in chapter five and the results from Appendix F with Baker's example #1 (reference [33], page 207).

requiring exponential time and/or space for solution. [36]. We will return to a related problem of this type in chapter five.

A possible alternative to the ranking algorithms discussed is Lin and Kernighan's heuristic solution to the traveling salesman problem. Rather than using costly algorithms to achieve optimal solutions, their heuristic technique achieves near optimal solutions at a greatly reduced cost. Their heuristic incorporates a mixture of look ahead, nearest neighbor, branch and bound, as well as backtracking techniques. Although only limited testing had been achieved on the heuristic, they reported statistically significant success in achieving global solutions. However, these results were only for the smaller size problems for which optimal solutions were known. [37]. Even though heuristic methods may not guarantee an optimal solution for the sequencing problem, they may have better cost/benefit ratios than known algorithms and as such invite serious consideration. It remains to be shown, however, whether Lin and Kernighan's heuristic for the traveling salesman problem is adaptable to the document ranking problem.

Excluded from this study are those factors which could affect the patrons' evaluation of retrieved material caused by:

1) The question - negotiation process used to determine the patron's information need.
2) The indexing strategy used to assign terms to the document.
3) Any factors pertaining to the method for delivering either the bibliographic citations or the documents to the patrons.

In concluding this chapter, two recent monographs in the field are noted, which have sharply signalled the need for clarification of the term "information" as well as the techniques fostered for its study.

Emerging from the 1972 NATO conference (Pittsburgh, Pennsylvania) Debons raises the following questions about the nature of information:

1) What is the representational process wherein information generation is optimized?

2) What cognitive mechanisms permit the optimum utilization of information?

3) How do information systems (environments) function to achieve optimum utilization?

4) What methodologies differentiate underlying mechanisms in information processing? [38, p. 473].

In addition, while developing models of the information transfer process, Kochen emphasizes, "there were hopes that information theory would help us encode descriptors for data bases ... that linear programming might help us design search strategies. This has not yet happened ..." He continues, "... it is possible that priorities of concern with file organization and data structures will be reversed toward more stress on what it takes to help the problem solver rather than how to optimally utilize computers and their programs." [9, pp. 4-5].

Focussing on Debons' fourth point, the methodology for optimal ranking of documents will be developed. Rather than deriving from the domain of linear programming as anticipated by Kochen, a dynamically programmed integer solution will result.

The "strategy" to be followed in this report will continue in chapter two by laying the groundwork for this study and developing an appropriate measure of retrieval effectiveness. In addition two* ranking

---

*Since the effectiveness of these rules will be established by comparing them with the optimal ranking rule, in reality there are three ranking procedures. The latter procedure, however, is not formally developed until chapter five.

procedures, the probability ranking rule and the optimal probabilistic strategy, will be inspected as likely candidates for "optimal" ranking. Chapter three will identify limits for which probability ranking is optimal.

Quantifying the magnitude of the suboptimality of simple probability ranking, a simulation model of the problem is developed in chapter four and used to present some empirical results.

The optimal ranking rule will be developed in chapter five. Since the algorithm is a member of a class of computationally difficult problems, its efficiency will be analyzed in this light.

Developing an appropriate cost/benefit criterion, in chapter six the break even performance point will be identified between the optimal ranking rule and the probability ranking rule.

Finally, chapter seven will summarize the study.

Chapter 2

A SEARCH FOR AN OPTIMAL RANKING RULE

To investigate the effect of probability theory and sequential search theory on document ranking and also lay the groundwork for development of an optimal ranking rule, it is necessary to define a model of a theoretical retrieval system (RS). (A generalized RS model will be developed here even though a restricted version will later be used for ease in presenting several counter examples.) Assume that several patrons have submitted the same request (although their information needs, conceived as psychological states, may be different) and suppose also that they have reported utility* estimates (at least the initial RS users) for all documents. Envision the initial set of patrons using the RS as representing an exhaustive set of the kinds of patrons who would subsequently interrogate the RS with the same or similar request. The utility estimates made by the initial users of the system may later be modified, but essentially these estimates form the basis for future document retrieval and ranking. (When the expressions "patron-types" or "user-types" are subsequently used, these terms will refer to users of the retrieval system for whom no utility estimates may exist, but whose

---

*Utility is regarded in the sense proposed by Cooper [39]. As the patron encounters a document he will estimate the utility which that article has to him. If he feels benefited from the experience, then he is likely to assign it a positive value. If, on the contrary, he feels that his time has been wasted, then he may wish to penalize the RS by assigning the article a negative utility.

information needs or psychological states are represented by at least one of the initial users of the retrieval system.) Assuming equivalency between the initial users and the user-types requires some scheme for obtaining and recording the initial users' utility estimates for all documents and requests. The success of the RS model is contingent upon the scheme used to gather and utilize these estimates.

The Retrieval Rule

Fundamental to the RS design is its retrieval rule. In the present model the retrieval rule has the role of both retrieving a set of documents and ranking these in some useful sequence. Ranking (on the average) minimizes the necessity to search negative or low utility documents among the retrieved set. Disagreement among the users in the ranking given to a document is not necessarily due to poor performance of the retrieval rule, but rather to the likelihood that dispute exists among the users as to its utility. The utility assessments assigned to the retrieved documents may differ for each patron because of differing information needs. The retrieval rule simply attempts to rank the retrieved documents so that the patron will on the average experience maximum utility.

The Stop-Search Point

Discussion of document ranking also requires that a "stop-search" point be defined for each patron if the ranked output created by the retrieval rule is to have any benefit. If all RS users read the entire retrieved set of documents--even though they are ranked in a beneficial way, the benefit due to ranking is negated. Therefore, each patron-type

must have a predetermined threshold level or an achieved utility* value which when attained the patron ceases to consider any more retrieved documents. Perhaps the achieved utility is a "utility density" function. That is, as the rate of high satisfaction documents drops significantly or the satisfaction estimates assigned become negative, then the patron-type stops his search.

A Measure of Retrieval Effectiveness

Restricting the generality of the preceding definition of utility will simplify both the visual representations and the performance criterion to be used in the comparisons that follow. In this case document utility estimates are replaced by relevance judgments.** All documents judged relevant by the initial patrons have equivalent value and all documents judged not relevant have the same negative value. The stop search point, under the restriction, is defined such that all users continue scanning the ranked retrieved set until they have encountered all relevant documents corresponding to their user-type. Since the stop-search point may involve different numbers of relevant documents for different patron-types, an appropriate measure of performance accommodating these variations may be computed by tallying the average

---

*It is important to emphasize that "achieved utility" does not imply that the patron can receive more than the RS is capable of delivering. The retrieved material may not satisfy the patron in a complete sense since what he needs may simply not exist in the RS. In addition, since none of the retrieved output may be of value to him, his stop-search point may be due entirely to frustration.

**In general, utility estimates assigned to a document will be affected by situational variables (eg., time, place, mood of user) as well as whatever influence previously scanned documents may have had upon the patron. Documents that have been judged relevant, however, will tend to remain so, even if offering no new insights to the user.

number of nonrelevant documents encountered by each user--or, more simply, the total number of nonrelevant documents encountered by all users. Transforming the performance measure from utility to relevance requires that the optimizing criterion be adjusted from "select the sequence which maximizes the users' average utility" to "select the sequence which minimizes the average number of nonrelevant documents encountered by the users."

Thus, in summary, documents are retrieved and ranked on the basis of relevance judgments obtained from a prior set of users. All patrons who have submitted the same information request (although their information needs may differ) are assigned to the same group, and their diversity of information needs are assumed to represent an exhaustive set of "patron-types" who will use the RS in the future. In addition, all patron-types exit from searching the document sequence when (and only when) they have achieved their a priori stop-search points--manifest by their encountering all relevant documents associated with their patron-types.

Testing the Ranking Rules

Two retrieval rules will be used to illustrate the consequence of sequential search theory and contrast the effect of document ranking on RS performance. The first will be called the probability ranking rule (PRR). It is based upon the frequency interpretation to probability theory and (analogous to sequential optimization in search theory) the principle that all "items should be arranged in decreasing order of the probability that an item will yield the information sought." [26, p. 131] According to this theory each user-type would achieve its stop-search

point with greatest efficiency by receiving that document judged relevant for the greatest number of patron-types first; the document judged relevant for the next greatest number of patron-types second; and so forth. The last document assigned to the suggested searching sequence would have been judged relevant for the least number of patron-types. Conceivably, this last document could have been rated as having high value to one user, but when considering its overall value, as determined by all users, the net effect would be to assign it to the last rank.

The other retrieval rule used for comparison will be called the optimal ranking rule (ORR). It selects a sequence for which the patron-types will encounter the fewest number of nonrelevant documents while, collectively, the types are achieving their stop-search points. (The development and evaluation of the ORR constitutes a major part of the research reported here, but will not be described until chapter five.)

To test the effectiveness of the two retrieval rules, the following hypothesis is proposed:

HYPOTHESIS:

> If a document retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of usefulness to the users who submitted the request, where the probabilities are estimated correctly on the basis of whatever data have been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data. [40, p. 4].

A Counter Example to the Optimality of the PRR

For a set of users having the same request let their relevance judgments be represented in the document-user matrix of Figure 2-1. An "X" in a location signifies that the i^{th} document is judged relevant

by the j^{th} user. Inasmuch as the documents are ranked in decreasing number of relevance "votes", the implied search strategy under the PRR would be for the user to begin by considering first document #1 and then continue selecting subsequent material in the downward order depicted by the row sequence of the matrix. The stop-search point is reached when each user identifies his last relevant document.

Users

Documents / Row	1	2	3	4	5	6	7	8	9	10	Row
1	x	x	x	x	x	x	x				1
2	x	x	x	x	x	x					2
3	x	x	x	x	x						3
4	x	x	x	x							4
5					x	x	x				5
6								x	x		6
7										x	7

A DOCUMENT RANKING ACCORDING TO THE PRR

Figure 2-1

From the matrix in Figure 2-1 one observes that the stop-search points of the user-types may vary, since their stopping points range from one to four relevant documents. Users one through four encounter no nonrelevant documents, user five encounters one, user six encounters two, user seven encounters three, users eight and nine encounter five each, and user ten encounters six nonrelevant documents before reaching their stop-search points. The accumulative total of nonrelevant documents encountered by all users would be 22, or an average of 2.2 per

patron. Since this statistic is a measure of the average difficulty* each user experiences while achieving his stop-search point, it seems appropriate to call it the average search length. The strict decreasing likelihood of achieving a relevant document as portrayed in this example is in accordance with "distributing the effort sequentially so as to maximize the probability of target detection with effort expended thus far." [25, p. 332].

Next, consider the counter example to the PRR as represented by the permuted row sequence displayed in Figure 2-2. The search length for this sequence is 17, an average of 1.7 nonrelevant documents per user, or better than 22% decrease in the average number of nonrelevant documents encountered by the users as they achieve their respective stop-search points.

It seems surprising that the PRR would yield a suboptimal result. Is this observation atypical of normal expectation? On the other hand, other factors remaining the same, if a RS consistently delivered results at 20% less effort to the patron, its services would, clearly, be more desirable. Although the preceding counterexample does not establish consistent failure of the PRR in achieving "overall effectiveness of the system to its users", from a theoretical viewpoint, the optimality of the PRR is no longer "intuitively obvious". To further explore the counter example of Figure 2-2, several questions in the form of objections will now be pursued.

---

*"Difficulty" in this context occurs only while encountering nonrelevant material. Any inconvenience, irritation, or discomfort experienced in the perusal of relevant material is not apparent in this measure.

Users

Documents		1	2	3	4	5	6	7	8	9	10		
	6								x	x		1	
	5					x	x	x				2	
	1	x	x	x	x	x	x	x				3	
	2	x	x	x	x	x	x					4	Row
	3	x	x	x	x	x						5	
	4	x	x	x	x							6	
	7										x	7	

DOCUMENT RANKING PRODUCED BY THE ORR

Figure 2-2

Objection #1

Since the user-types in Figures 2-1 and 2-2 require different numbers of relevant documents to reach their stop-search points, shouldn't the user-types (matrix columns) be weighted to reflect their different stop-search points?

Response

Weighting the user-types would be desirable if they represented unequal numbers of system users, but any attempt to weight the documents for any user-type is tantamount to differentiating the value of one relevant document over another. To do this presupposes more information than is presently available in the user-document matrix and the restricted version of the RS model. In addition, consider the user-document matrix of Figure 2-3 in which all user-types have a stop-search point equivalent to obtaining the same number of relevant documents. Under the PRR their

search length is 27 nonrelevant documents--an average of 3 per user.

Users

Documents	1	2	3	4	5	6	7	8	9	Row
1	x	x	x	x	x					1
2	x	x	x	x	x					2
3						x	x	x	x	3
4						x	x	x	x	4
5	x			x			x			5
6		x			x			x		6
7			x			x			x	7

DOCUMENT RANKING ACCORDING TO THE PRR

Figure 2-3

Users

Documents	1	2	3	4	5	6	7	8	9	Row
6		x			x			x		1
2	x	x	x	x	x					2
1	x	x	x	x	x					3
5	x			x			x			4
7			x			x			x	5
3						x	x	x	x	6
4						x	x	x	x	7

A COUNTER EXAMPLE TO THE PRR WHERE EACH USER HAS EQUAL STOP-SEARCH POINTS

Figure 2-4

Permuting the document sequence of Figure 2-3 while maintaining the same relevance judgments per user resulted in an improved search length of 20, as shown in Figure 2-4. The latter requires 26% fewer nonrelevant documents on the average than its frequency ranked companion. Thus a counter example also exists to the optimality of the PRR when all user-types have the same stop-search points.

Tacitly assumed in the examples is that five nonrelevant documents encountered by one patron is equivalent to one nonrelevant document encountered by each of five patrons. Objections to this point, however, ultimately stem from the inadequacy of binary relevance judgments, since using the dichotomous values implies that all relevant documents are equivalent and that lost effort, discomfort, etc., arising from each nonrelevant document is the same for all users.

Although using the restricted version of the RS model has simplified the demonstration of counter examples, similar counter examples to probability ranking could be generated when the more general notion of utility is used.

Objection #2

The documents are ranked by numbers of relevance votes or user-type assessments whereas the optimizing criteria is specified in terms of "nonrelevant documents encountered by the users." Although related, the ranking and optimizing criteria are, nonetheless, not the same. This may account for the discontinuity displayed by the counter examples, Figures 2-2 and 2-4.

Response

That the two criteria are different is true. However, they are

equivalent to each other through an additive constant. Let X_i equal the probability that the i^{th} document is relevant to the user-types. For N documents the ranking criterion has been

$X_1 \geq X_2 \geq \ldots \geq X_i \geq X_{i+1} \geq \ldots \geq X_N$. The optimizing criterion on the other hand has been $(1-X_1) \leq (1-X_2) \leq \ldots \leq (1-X_i) \leq (1-X_{i+1}) \leq \ldots \leq (1-X_N)$. Multiplying each expression by a negative one reverses both the sign of the X_i and the direction of the inequality. Finally, adding 1 to each expression transforms the second expression into the first. Since these arithmetic operations have not altered the order of any of the X_i's, minimizing the probability of nonrelevant documents at each level is equivalent to maximizing the probability of relevant documents at each level. Thus, in the restricted version of our RS model the ranking and optimizing criteria are equivalent.

Objection #3

The probability ranking rule is computed strictly on the basis of comparing row totals of the document-user matrix. This a priori use of statistics does not permit any but the most simplistic version of decision making. Shouldn't the ranking criterion be based upon an optimal procedure which maximizes the expected utility or minimizes the expected loss of the system users. The optimal ranking rule seems to take advantage of all information in the user-document matrix, simultaneously. Shouldn't an optimal probabilistic strategy (OPS) be selected that does the same?

Response

An OPS based upon minimizing expected loss would yield an equivalent* result when compared with the ORR. But even in the simplified case where all users achieve their satisfaction points after obtaining only one relevant document, an optimal probabilistic strategy would require exponential effort of N! magnitude. A more complex version can be developed to represent the case in which patron types have variable satisfaction levels, but consider the following simple case in which all users are after only one relevant document.

Assume the existence of a document-user matrix representing the various information needs of all users who have submitted the same information request to a retrieval system. To minimize the average search length (SL) for N documents and M users over all document orderings, let D_1= the document assigned to the first position within the search sequence, D_2= the document assigned to the second position, ...,D_N equals the document assigned to the last position. Recalling the convention that each nonrelevant document contributes a numerical value of one to the search length, requires

$$\min(\overline{SL}) = \min_{\text{all document permutations}} \Big[(0)P(d_i=D_1 \cap rel) + (1)P(d_j=D_2 \cap rel \cap d_i=D_1 \cap \overline{rel}) + (2)P(d_k=D_3 \cap rel \cap d_i=D_1 \cap \overline{rel} \cap d_j=D_2 \cap \overline{rel}) + \ldots + (N-1)P(d_m=D_N \cap rel \cap d_i=D_1 \cap \overline{rel} \cap d_j=D_2 \cap \overline{rel} \cap \ldots \cap d_\ell=D_{N-1} \cap \overline{rel}) \Big]$$

*"Equivalent" is used rather than "same" since numerous sequences may exist at which expected utility is maximized (or expected loss is minimized).

Observing that all probabilities are a function of D_1, all but one a function of D_2, etc.,

$$\min(\overline{SL}) = \min_{(d_i)} \Big[(0)\, P(d_i = D_1 \cap rel)$$

$$+ \min_{(d_j \neq d_i)} \Big[(1) P(d_j = D_2 \cap rel \,|\, d_i = D_1 \cap \overline{rel}) P(d_i = D_1 \cap \overline{rel})$$

$$+ \min_{(d_k \neq d_i, d_j)} \Big[(2) P(d_k = D_3 \cap rel \,|\, d_j = D_2 \cap \overline{rel} \cap d_i = D_1 \cap \overline{rel})$$

$$P(d_j = D_2 \cap \overline{rel} \,|\, d_i = D_1 \cap \overline{rel}) P(d_i = D_1 \cap \overline{rel}) + \ldots$$

$$+ \min_{(d_n \neq d_i, d_j, \ldots, d_{n-1})} \Big[(N-1) P(d_n = D_N \cap rel \,|\, d_m = D_{N-1} \cap \overline{rel} \cap \ldots$$

$$\cap d_j = D_2 \cap \overline{rel} \cap d_i = D_1 \cap \overline{rel}) P(d_m = D_{N-1} \cap \overline{rel} \,|\, d_\ell = D_{N-2} \cap \overline{rel} \cap \ldots$$

$$\cap d_j = D_2 \cap \overline{rel} \cap d_i = D_1 \cap \overline{rel}) \ldots$$

$$P(d_j = D_2 \cap \overline{rel} \,|\, d_i = D_1 \cap \overline{rel}) P(d_i = D_1 \cap \overline{rel}) \Big] \ldots \Big] \Big] \Big]$$

Omitting the (0) component in the above and factoring the remaining expression results in,

$$\min(\overline{SL}) = \min_{(d_i)} \Big[(0) + \min_{(d_j \neq d_i)} \Big[(1) P(d_j = D_2 \cap rel \,|\, d_i = D_1 \cap \overline{rel}) +$$

$$\min_{(d_k \neq d_i, d_j)} \Big[(2) P(d_k = D_3 \cap rel \,|\, d_j = D_2 \cap \overline{rel} \cap d_i = D_1 \cap \overline{rel}) + \ldots$$

$$+ \min_{(d_n \neq d_i, d_j, \ldots d_{n-1})} \Big[(N-1) P(d_n = D_N \cap rel \,|\, d_m = D_{N-1} \cap \overline{rel} \cap \ldots$$

$$\cap d_j = D_2 \cap \overline{rel} \cap d_i = D_1 \cap \overline{rel}) \Big]$$

$$P(d_m = D_{N-1} \cap \overline{rel} \,|\, d_\ell = D_{N-2} \cap \overline{rel} \cap \ldots \cap d_j = D_2 \cap \overline{rel} \cap d_i = D_1 \cap \overline{rel}) \Big] \ldots$$

$$P(d_j = D_2 \cap \overline{rel} \,|\, d_i = D_1 \cap \overline{rel}) \Big] \; P(d_i = D_1 \cap \overline{rel}) \Big]$$

The optimal strategy is achieved when the document ranking D_1, D_2, D_3, ... , D_N has been assigned documents appropriate to min (SL). The above expression is a series of N-1 multiplications, and since D_1 can be assigned N documents, D_2 N-1 documents, ...,D_{N-1} two documents, there exist (N)(N-1) ... (2) = N! unique sequences needing to be checked. $N! \approx N^N$, which is a high order exponential, akin to a brute force evaluative approach, and a computational burden to avoid. This problem will reappear in chapter five.

Several assumptions pertaining to the availability of a priori data have been made or implied while developing the previous ranking rules. The intent of the following objections will be to question these assumptions in order that the sources of data for and applications of any ranking criteria based upon a priori data will be better understood.

Objection #4

Since it is unlikely that a patron assigning utility estimates to a retrieved set of documents will submit the same request again, of what use is a search strategy based upon his utility estimates? Such a retrieval system appears to have the patron requesting, evaluating, and then repeating the request, which is akin to a treadmill of endless torture.

Response

Generally, an optimal search strategy would not be useful if a patron could benefit only by resubmitting the same request. In fact, if a patron were to repeat his request, assuming he is able to recall the results of his previous efforts, or a patron could be identified as

"equivalent" to a previous patron for whom document utility estimates were known, his best search strategy would be to select from the retrieved documents only those that had been found useful by that patron or type of patron. In such a case the optimal search strategy for all users is not necessarily the best one, since the specific utility assignments are already known.

However, as in the SDI example in chapter one and the RS model described earlier in this chapter, consider the situation in which a first time request is submitted for which document relevance estimates have been acquired from previous patrons for the retrieved documents. If we are fairly confident that the information need of the patron submitting the request is represented by one of the previous patrons, but are uncertain as to which one, the optimal ranking rule would yield the best search strategy for him. (In the RS model the "previous patrons" have been categorized into patron or user-types.) Maintaining the equivalent effect that each document would have in ranking the retrieved documents necessitates adjusting for the different numbers of requestors associated with each user-type. In this case a weighting factor, possibly based upon the number of requestors associated with each type would be multiplied times the utility estimate assigned each document in the respective user-type (column).

For another example, consider the task performed by a group of reference librarians. Numerous reference works are routinely consulted in an attempt to locate desired information. The answer sought may be contained in one or several sources. Or the point of interest may not be explicity recorded but only implied by garnering bits of inferential data from many sources but less than the entire retrieved set. Again, if

the individual knows what he wants, reassemblage of that set of relevant material represents his optimal search strategy. However, if the uninitiated or untrained individual who is learning to use the reference tools has similar utility estimates as a person or as a class of persons for whom estimates have been obtained, the optimal ranking rule based upon those estimates would supply the best search strategy for him.

A third example could be illustrated through the procedure used to "establish" a bibliographical entry. When ordering and cataloging new titles, it is imperative that librarians agree on the type and form of the bibliographic entry to the title. Establishing the authority form of, for example, the main entry to the title may require the use of several bibliographic sources. The sleuthing technique used and the sequence in which the reference sources are searched vary from one searcher to another, but an "optimal search strategy" would be identified once the best rationale was established for making the appropriate technical decisions. Such a strategy would be useful for instructional purposes in helping trainees to differentiate the nature of various reference sources and compare the search heuristics used by their tutor-types--thus sharpening their own proficiency. [41, p. 160].

The elements necessary for the proper application of the optimal ranking rule in the foregoing examples are:

1) The user population, the nature of user requests, and the retrieval system data base must be fairly stable. A large differential in composition of the user population could alter the utility estimates of the "previous users." If user requests have no repetitiousness, the usefulness of any ranking rule diminishes. If the data base is in a state

of flux, so will be the retrieved set of documents for a given request, and the accuracy of utility estimates, used for maximizing future performance, would be questionable.

2) Dispute or difference of opinion must exist among the users as to the utility of individual documents or decisons to be made and, consequently, the order in which they should be read or sequenced. (If conflict does not exist, then the results to be shown in chapter three suggest that the probability ranking rule will be optimal.) Differences, for example, could arise if several sources, each, contain the desired information, but vary in the ease with which the user can obtain or understand them. In addition, the patrons may have differing backgrounds and preparation which affect relative ease of document use. Or different applications of the information may affect the utility estimates assigned by the patrons.

3) Accurate feedback from the patrons in the form of document value or utility estimates must be available. Data collection costs for obtaining this information might be a deterrent, but sampling techniques could be used to hold costs in line. The method of interrogation used to obtain the estimates would be crucial to their validity. Minimizing the disturbance of the patron from his normal mode of search would be essential. An approach to obtaining user estimates involves the use of role playing. Participants could be instructed to "imagine" ten different types of users who would submit the same request.

Each user would be portrayed as having a different emphasis on his request, background knowledge, etc. The participant is asked to rank the retrieved set (or reference collection) with respect to its value to each imagined user. The results from role playing would be analyzed to identify homogenous groups among the user population.

Objection #5

Application of the optimal ranking rule as illustrated in Figure 2-2 seems to penalize the majority of users who are seeking larger numbers of documents on the basis that a small minority of users, who wish only one or two relevant documents, disagree with them. The end result is that the majority of people have more nonrelevant material to search through while the minority have less.

Response

Establishing an analysis such as this on the basis of one matrix is questionable. By contrast, the majority of users in both Figures 2-4 and 5-2 have fewer nonrelevant documents to search while the minority of users have more. If one thinks in terms of averages, the ORR will always yield a result that is less than or equal to the average number of documents one would search under the probability ranking rule. If the patron knows which material is relevant to him, then he should pursue it directly and thus surpass the minimum average inconvenience set forth by the optimal procedure. If the patron doesn't know which material is relevant, then neither does he know whether or not he is working more or less to reach his stop-search point--except that in the long run he will work less by following the optimal search strategy.

Objection #6

All of the ranking rules used thus far presuppose that for a given request every user knows in advance the amount of relevant material necessary to satisfy his information need. Is this realistic?

Response

The ranking rules function on the basis that a priori information is known about the set of users. Given this information the rules attempt to exploit the fact that almost all users will be satisfied before reading all the retrieved documents and when satisfied they will exit from the search sequence. The PRR, for example, attempts to put the more relevant material at the beginning of the search sequence and the less relevant material at the end. If the user population is accurately represented by the a priori utility estimates, the ORR on the other hand maintains that the users should be satisfied with minimum average search effort. If patrons wish to continue their search past their normal "stop-search" point (which is permissable), or the patrons do not have a stop-search point, or no a priori data is available to lay the basis for a search strategy, then any benefit of having a ranking rule is negated.

Returning to the hypothesis at the first part of this chapter, the counter examples presented have shown that PRR (and consequently the simplified model of sequential search theory) does not provide the "best that is obtainable" document ranking. However, it is possible that the observed contradictions may simply be isolated anomalies and in general the PRR will provide an "overall" effective method for imposing a search strategy. This question, specifically, will be addressed in chapter four. In chapter three, however, conditions for the suboptimality and optimality of the PRR will be presented.

Chapter 3

THE LIMITS OF PROBABILITY RANKING

In chapter two, several counter examples to the optimality of probability ranking were identified. In this section the conditions for optimal ranking to exist under the probability ranking rule will be derived. Three cases will be investigated. The first assumes that the stop-search points for all patrons will be climaxed after each has located one relevant document. This proof follows that of Klonk [42, pp. 8-9] and Baker [33, pp. 204-5]. Next, a more general case is presented for the situation where each user's stop-search point may include any number of relevant documents. (These cases are presented under the restriction that documents are rated either relevant or nonrelevant.) Finally, the results for the case in which only two documents are retrieved will be analyzed because of special interest.

Case I: Stop-Search Point Equivalent to One Relevant Document

Let Q_j equal the set of users who are satisfied by the j^{th} document and by no other document. It is assumed that all users will achieve their "stop-search" point after obtaining one relevant document. Hence

$$Q_j \cap Q_k = \emptyset \quad \forall k \neq j$$

so that the Q's are disjunctive sets.

With the number of users belonging to Q_j represented by $|Q_j|$ the total search length as defined in the previous chapter becomes (where the

number of documents equals N):

$$SL = \sum_{i=1}^{N} |Q_i| (i-1) \qquad (1)$$

The probability ranking rule imposes upon the sets Q_j the condition that the cardinality (i.e., the number of users) of these sets be decreasing. That is,

$$|Q_1| > |Q_2| > |Q_3| > \cdots > |Q_N| \qquad (2)$$

The assertion is that ranking the documents in decreasing order of popularity to the users will be optimal in that the search length (i.e., the number of nonrelevant documents encountered by the users) will be minimal.

PROOF:

Assume that an improved (i.e., shorter) search length can be obtained by breaking from decreasing probability of relevance. That is, assuming that in one instance $|Q_i| < |Q_{i+1}|$ it is proposed that the search length for this sequence, having an inversion of two adjacent documents, will be less than the search length associated with the same documents arranged in strict decreasing popularity to the users. Applying equation (1) the search length for the sequence ranked according to the PRR is*:

$$SL' = |Q_1|(0) + |Q_2|(1) + \ldots + |Q_{i+1}|(i-1) + |Q_i|(i) + \ldots + |Q_N|(N-1) \qquad (3)$$

---

*Throughout this presentation a prime (i.e., SL′)will always be associated with the document sequence resulting from the PRR--a ranking possessing decreasing numbers of relevance "votes" for each subsequent document.

The search length for the alternate sequence is:

$$SL = |Q_1|(0) + |Q_2|(1) + \ldots + |Q_i|(i-1) + |Q_{i+1}|(i) + \ldots + |Q_N|(N-1) \qquad (4)$$

The sequence of documents in both arrangements is identical except that the i^{th} and i+1 documents have been interchanged in (4) in opposition to the PRR's requirement of decreasing cardinality of the Q's. Subtracting equation (4) from (3), we obtain

$$SL'-SL = |Q_{i+1}|(i-1) + |Q_i|(i) - \left[|Q_i|(i-1) + |Q_{i+1}|(i)\right] = |Q_i| - |Q_{i+1}| \qquad (5)$$

Since by definition $|Q_i| < |Q_{i+1}|$, $SL'-SL<0$. Thus, $SL>SL'$ which contradicts our first assumption in the proof and shows that the ranking associated with SL´ is better than the alternate ranking corresponding to SL. (Had two nonadjacent documents been switched in the proof--for example, Q_i and Q_{i+k} where $|Q_i|<|Q_{i+k}|$, both terms in the final expression of equation (5) would be multiplied by k. But the sense of the inequality in equation (5) would not be changed.) Hence, based upon the PRR, ranking the documents by their decreasing popularity to the users is optimal for the case in which each user has a stop-search point equivalent to one relevant document.

Case II: Unrestricted Stop-search Point

Now consider the general case in which the system users may each have any number of relevant documents associated with their stop-search points. It is assumed that when relevant material is encountered by each user, he will recognize it as such, and that his search will continue until all relevant material, consistent with his stop-search point, is obtained. This situation follows that of the RS model described in chapter 2--for which counterexamples to the PRR are known to exist. To

compare optimal document ranking with that produced by the PRR, a measure of performance similar to equation (1) but adapted to this more general case, must first be defined.

The search length equivalent to equation (1) for this general case becomes more complex since the relevant material sought by a user may be any combination of the documents within the sequence. For M users and N documents equation (1) now becomes:

$$SL = \sum_{i=2}^{N} \left[\sum_{j=1}^{M} | Q_{i,j} | (i-1) - \sum_{j=1}^{M} | Q_{k_j,j} \cap Q_{i,j} | (k_j) \right] \qquad (6)$$

$$\text{Where } k_j = \left[\begin{array}{l} \underline{\max}\ (k) : Q_{k,j} \cap Q_{i,j} \neq \emptyset \quad \forall k < i \\ 0, \text{ if } Q_{k,j} \cap Q_{i,j} = \emptyset \quad \forall k < i \end{array} \right]$$

The first expression enclosed in the brackets on the right, which is similar to the previous search length equation (1), computes the search length to the i^{th} document (for each of the users) as if there were no preceeding relevant material in the sequence. That is, assuming that the i^{th} document is relevant to the j^{th} user the expression treats him as if he has just encountered (i-1) nonrelevant documents. The second expression on the right of equation (6) adjusts the first expression for those instances where relevant documents in the sequence have previously been encountered by any user.

For example, assuming that the k^{th} and i^{th} documents in the sequence (k<i) were judged relevant by the j^{th} user, equation (6) determines the search length to the k^{th} document (i.e., k-1, since there are no preceding relevant documents the coefficient of the second expression equals zero) and adds this amount to the difference (i.e., i-1-k) between the search lengths of the k^{th} and i^{th} documents--with the result that this

user encounters i-2 nonrelevant documents while attaining the two relevant ones. Thus, for each user the amount of nonrelevant material encountered between the relevant documents is accumulated, according to equation (6), as he evaluates the sequence. The condition on k_j associated with equation (6) simply restricts the row value, it may assume, to be equal to the most recent relevant document previously encountered in the sequence by the j^{th} user. (Since the value of k_j is equivalent to the position of the previous relevant document, its value may vary with each user. Hence, the subscript j has been added to indicate that these values are user dependent.)

Although appearing somewhat awkward, the formulation of equation (6) is essential in order to accommodate the variable stop-search points of the users while determining their search lengths. The one condition under which the expression would fail to give a true count of the number of nonrelevant documents is for the case where (for a given user) all of the documents are nonrelevant. But since our ultimate objective is to determine the document sequence that minimizes the average search length to the users, any user for whom no benefit exists, whatsoever, would feel the same regardless of how the document sequence were permuted and consequently would have no influence in determining the optimal sequence. In this instance the optimal order would not be affected, but the average search length for each user should be increased by the amount N/M for each occurrence of the problem. Since the development of the proof for Case I, however, involved the difference between two search lengths, the effect due to the problem just stated was cancelled out--and thus, could be ignored. An example of how equation (6) was used to determine the search length of the document sequence in the

user-document matrix of Figure (2-4) may be found in Appendix D.

Having obtained the expression for search length and following the procedure established in the Case I proof, we will now attempt to show that decreasing probability of relevance to the users will again yield the optimal search strategy for the situation where the system users, now, may have any number of relevant documents associated with their stop-search points. Assuming that all $\sum_{j=1}^{M} |Q_{s,j}|$ are cardinally decreasing for s = 1, 2, ..., N except for two terms where

$$\sum_{j=1}^{M} |Q_{i,j}| < \sum_{j=1}^{M} |Q_{i+1,j}| ,$$

two search sequences, one representing the PRR, the other having the ith and i+1 documents interchanged, will again be constructed and then the difference between the two obtained. The details involved in obtaining the resulting expression, equation (7), are presented in Appendix E. It should be emphasized that equation (6) is a general expression--true for any document arrangement. Equation (7) on the other hand is true only when obtaining the difference between two search lengths--one search length for a sequence ranked by the PRR and the alternate for a similar sequence which has two adjacent documents interchanged.

$$SL'-SL = \sum_{j=1}^{M} |Q_{i,j}| - \sum_{j=1}^{M} |Q_{i+1,j}|$$

$$+ \sum_{\ell=i+2}^{N} \sum_{j=1}^{M} \left[|Q_{\overline{k}_j^{\leq \ell-1},j} \cap Q_{\ell,j}| \left(\overline{k}_j^{\leq \ell-1}\right) - |Q_{\overline{k}'^{\leq \ell-1}_j,j} \cap Q_{\ell,j}| \left(\overline{k}_j^{\leq \ell-1}\right)' \right] \qquad (7)$$

where k_j = MAX (p) : $Q_{p,j}$ = relevant, p=i, i+1

and ℓ= MIN(s) : $Q_{s,j}$ = relevant, $s \geq i+2$

Equation (7) holds the same purpose for Case II as equation (5) did for Case I. In the former case equation (5) established a sufficient condition for the optimality of the PRR on the basis of two terms that are equivalent to the first two expressions on the right of equation (7). However, we may anticipate that the additional expression in equation (7) is going to cause a different interpretation than occurred in Case I.

Observing that for the PRR to be an optimal procedure in Case II, $SL'-SL$ must be less than zero. By definition, the net result of the first two expressions on the right is less than zero. With the addition of the third expression, however, a positive value may result which would be sufficient to offset the negative value obtained from the first two expressions--thus creating a counter example to the optimality of the PRR.

Analyzing the third expression of equation (7) we observe that it compares only the first relevant documents following the i^{th} and $i+1^{th}$ documents in the probability and optimal ranked sequences with relevant documents that may exist at the i^{th} and $i+1^{th}$ positions for each user. If there are no relevant documents for the j^{th} user at positions $\geq i+2$ or at either of the i^{th} or $i+1$ positions in the sequences then both $|Q_{k_{j,j}^{\leq \ell-1}} \cap Q_{\ell,j}^{\ell}|$ and $|Q_{k'^{\leq \ell-1}_{j,j}} \cap Q_{\ell,j}^{\ell}|$ will be empty ($\emptyset$) and the third expression will vanish for that user. Moreover, if both of the i^{th} and $i+1^{th}$ documents are relevant for the j^{th} user, then the maximum values of $\left(k_j^{\leq \ell-1}\right)$ and $\left(k_j^{\leq \ell-1}\right)'$ will be identical ($=i+1$), and the third expression will vanish even if the sets in the previous statement are not empty.

So, considering all possible user cases, the last expression of equation (7) will yield a positive value if fewer nonterminal relevant documents* are moved toward the end of their search sequences than terminal relevant documents are moved toward the beginning of their sequences (during the inversion of the i^{th} and i+1 documents). When this positive value is equal to or exceeds the negative value from $\sum_{j=1}^{M} |Q_{i,j}| - \sum_{j=1}^{M} |Q_{i+1,j}|$, a counterexample exists to the PRR.

The reason why the PRR produces suboptimal search strategies is that it is indifferent as to whether or not a given document in a sequence will be terminal or nonterminal for a given user. For example, if seven out of ten users judge a given document to be relevant, the probability statistic does not differentiate between the $\binom{10}{7} = 120$ different ways in which the 7/10 may be represented. If the document would be ranked highly by most of the users, but is nonterminal for them it may be moved towards the end of the sequence without severly penalizing the users' search lengths. Thus, determining a document's optimal position within a sequence cannot be done solely on the basis of the number of users who regard it to be relevant.

To illustrate the results of equation (7), consider the following three examples. Since the formulation of the proof is only valid for the case in which two consecutive documents among a cardinaly decreasing sequence are interchanged, the illustrations are contrived to demonstrate

---

*"Nonterminal relevant document" simply means that the user has not achieved his stop-search point at that document. On the other hand, a "terminal relevant document" is one for which a user achieves his stop-search point and will exit from his search.

Example #1: An Application of Equation (7)

User

Row	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x	x	x	x			
2	x	x	x	x	x	x				
3	x	x	x	x	x					
4	x	x	x	x						
5					x	x	x			
6								x	x	
7										x

$Q_{i+1,j}$
$Q_{i,j}$

(Cardinally Decreasing Sequence)

Figure 3-1

User

Row	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x	x	x	x			
2	x	x	x	x	x	x				
3	x	x	x	x	x					
4					x	x	x			
5	x	x	x	x						
6								x	x	
7										x

$Q_{i,j}$
$Q_{i+1,j}$

Figure 3-2

Applying equation (7) with i=4 for this example,

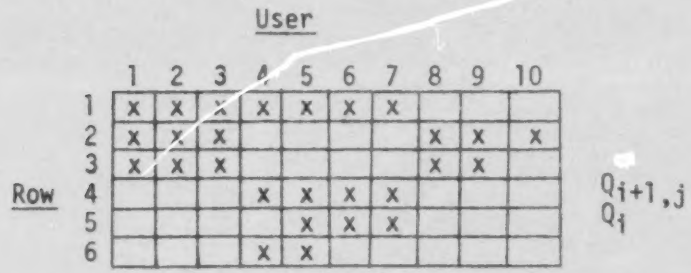
$$
\begin{aligned}
SL'-SL &= 3 - 4 \\
&+(0)(5)-(0)(4)+(0)(5)-(0)(4)+(0)(5)-(0)(4)+(0)(5)-(0)(4)+(0)(4)-(0)(5) \\
&+(0)(4)-(0)(5)+(0)(4)-(0)(5)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)^{*} \\
&+(0)(5)-(0)(4)+(0)(5)-(0)(4)+(0)(5)-(0)(4)+(0)(5)-(0)(4)+(0)(4)-(0)(5) \\
&+(0)(4)-(0)(5)+(0)(4)-(0)(5)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)^{*} \\
&= -1
\end{aligned}
$$

For this inversion, decreasing relevance as produced by the PRR, remains optimal.

---

*As there are no relevant documents in either the i^{th} or i+1 row for users 8, 9, and 10 the terms $\binom{\leq \ell-1}{k_j}$ and $\binom{\leq \ell-1}{k_j}'$ have been represented by (∅).

Example #2: An Application of Equation (7)



(Cardinally Decreasing Sequence)

Figure 3-3

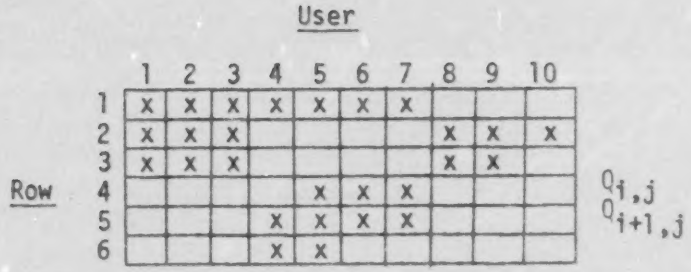


Figure 3-4

Applying equation (7) with i=4 for this example,

$$SL'-SL = 3 - 4$$
$$+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)+(1)(5)-(1)(4)+(1)(5)-(1)(5)$$
$$+(0)(5)-(0)(5)+(0)(5)-(0)(5)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)+(0)(\emptyset)-(0)(\emptyset)$$
$$= 0$$

For this inversion a counter example occurs since SL'=SL. However, the search lengths for both configurations are equal.

Example #3: An Application of Equation (7)

User

Row	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x	x	x				
2						x	x	x	x	x
3	x	x	x	x						
4							x	x	x	
5										x

$Q_{i+1,j}$
$Q_{i,j}$

(Cardinally Decreasing Sequence)

Figure 3-5

User

Row	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x	x	x				
2	x	x	x	x						
3						x	x	x	x	x
4							x	x	x	
5										x

$Q_{i,j}$
$Q_{j+1,j}$

Figure 3-6

Applying equation (7) with i=2 in this example,

SL´-SL = 4 - 5
+(0)(2)-(0)(3)+(0)(2)-(0)(3)+(0)(2)-(0)(3)+(0)(2)-(0)(3)+(0)(Ø)-(0)(Ø)
+(0)(3)-(0)(2)+(1)(3)-(1)(2)+(1)(3)-(1)(2)+(1)(3)-(1)(2)+(0)(3)-(0)(2)
+(0)(2)-(0)(3)+(0)(2)-(0)(3)+(0)(2)-(0)(3)+(0)(2)-(0)(3)+(0)(Ø)-(0)(Ø)
+(0)(3)-(0)(2)+(0)(3)-(0)(2)+(0)(3)-(0)(2)+(0)(3)-(0)(2)+(1)(3)-(1)(2)
= +3

In this case a counter example also exists. On the average there would be 3/10 less nonrelevant document encountered by the users through interchanging the second and third documents. End of examples.

a point. (However, to alter equation (7) such that nonadjacent documents are interchanged is a trivial process.) The search sequence followed by each user begins with the document represented by the top row of the matrix and proceeds downward. Information for the primed terms will be obtained from Figure (3-1), the cardinally decreasing sequence, while Figure (3-2) contains the data for the non-primed terms. The "x"es signify that the document is relevant to the corresponding user.

So far, the discussion has shown that ranking documents by decreasing usefulness to the users provides the optimal ranking when all users are satisfied with one and only one relevant document. However, when more than one relevant document is required by the users to achieve their stop-search points, ranking the documents by decreasing relevance --as provided by the PRR may not be optimal. A number of pertinent questions need to be raised.

1) If decreasing probability of relevance is not optimal, could one expect this condition to be a frequent occurrence? Or is the condition so infrequent that, for all practical purposes, it should be disregarded?

2) If the suboptimal condition does, (or can) occur frequently, to what extent is it likely to deviate from the optimal configuration? Is the relative difference in search length between decreasing probability of relevance and optimal ranking of sufficient magnitude to be of interest in the design of document providing systems?

Answers to these questions will be investigated in the next chapter. However, before proceeding it will be useful to obtain one more insight about the optimality of the PRR.

Case III: Only Two Documents Are Retrieved

Observing equation (6) we note that it is valid only when there are at least i+2 documents. In addition, by deleting the expression

involving the i+2 summation in equation (7), it becomes equivalent to equation (5) which established the condition under which the PRR is optimal. Since the minimum value of "i" is one and the third expression in equation (7) exists only at values $\ell \geq i+2$, we observe that, in general, the PRR is optimal when only two documents are ranked. For a given user, either one or two documents can be relevant. Only when three or more documents are sequenced can a potential counter example exist to probability ranking, and then only when the users need two or more relevant documents to achieve their stop-search points. This limiting factor of the optimality of the probability ranking rule will be of interest as we simulate the problem in the next section.

Chapter 4

SIMULATING PROBABILITY AND OPTIMAL RANKING

In the preceeding chapter a proof was developed to illustrate that under certain conditions counterexamples to the optimality of probability ranking would be obtained. The definition and analysis of the optimal ranking rule follows in chapter five, but in the present chapter computer simulation will be used to predict both the frequency of occurrence and the degree of degradation of the PRR's suboptimal performance.

The feature tending to be observed under a counterexample, such as Figure (3-6), is that conflict among the users, manifest by the lack of agreement in their relevance judgments, is present. In reality, variability in agreement among users does seem to be legitimate. Information for which the patron is already aware or for which he does not perceive its relevance to his need (even though at a later date it may become so) would not have high utility to him at the time and represents a realistic source of conflict with other system users. And, yet, portraying the users' relevance judgments in the counterexamples as having extreme distinctive patterns (i.e., mutually exclusive) or patterns that are identical seems to be atypical of reality. If we assume that user conflict is present, at what rate may we expect probability ranking to manifest a suboptimal search strategy? And when ranking by the PRR is suboptimal, to what extent is its search length

degraded? Empirical evidence will be generated via computer simulation to suggest answers to these two questions.

Simulating the Problem

The search strategy model which has formed the basis of discussion to this point is characterized as:

1) A set of patrons having a common request but differing information needs.

2) A stop-search point existing for each user at which he stops evaluating the ranked retrieved documents.

3) A sequential ordering of the retrieved documents (based upon prior knowledge of their value to the system users) which minimizes the average amount of nonrelevant material brought to the users' attention.

When it was determined that the probability ranking rule would not satisfy the third criterion above, an algorithm was developed that obtained the ordered sequence consistent with the minimum average search length. Since memory requirements of the solution algorithm are large, it was programmed in ALGOL so that string packing would be possible on a bit level. (Other languages could be used, but ALGOL was available.) The simulation program, of which a description follows was also programmed in ALGOL and run on the DEC KI/10 time share system at Brigham Young University.

Given a document-user matrix corresponding to N documents and M users, random numbers [43] were used to determine both how many documents and which documents were regarded "relevant" by each user. By randomly determining the users relevance assessments assurance is obtained that the results will not be based upon contrived user-document configurations as in the previous counterexamples. In addition, determining which documents are "relevant" for each user in this fashion

provides us with (simulated) a priori information about each user's stop-search point, since once the matrix is generated the "relevant" documents are known for each user.

In actuality neither the documents nor the users are real. The intent is to generate without bias numerous retrieved sets of documents that actual users might encounter, assume that actual users would make relevance judgments similar to our random simulator and then use these simulated retrieved sets of "documents" to test the degree of minimum search length degradation caused by the PRR.

Two statistics were recorded. First, the documents were sorted by decreasing number of relevance "votes" to represent probability ranking, and the search length was computed for all users. Next, using the optimal ranking algorithm, the minimum search length was computed for the same matrix. (The corresponding document sequence under the ORR was, also, noted.)

Results of the Simulation Experiment

Using "fresh" random numbers several hundred matrices were generated with numbers of users ranging from 2 to 40 and numbers of documents varying from 3 to 11. Combining the results from all simulated matrices, the search lengths associated with probability ranking agreed with the search lengths produced by optimal ranking 41% of the time. If a measure of agreement, A., is defined as:

$$A = \frac{\text{(No. times both procedures yield rank orderings having the same search length)}}{\text{(total No. Cases)}} \times 100 \qquad (1)$$

then Figure (4-1) indicates how A varies with number of documents, N, and also number of patrons, M. For simply a few retrieved documents

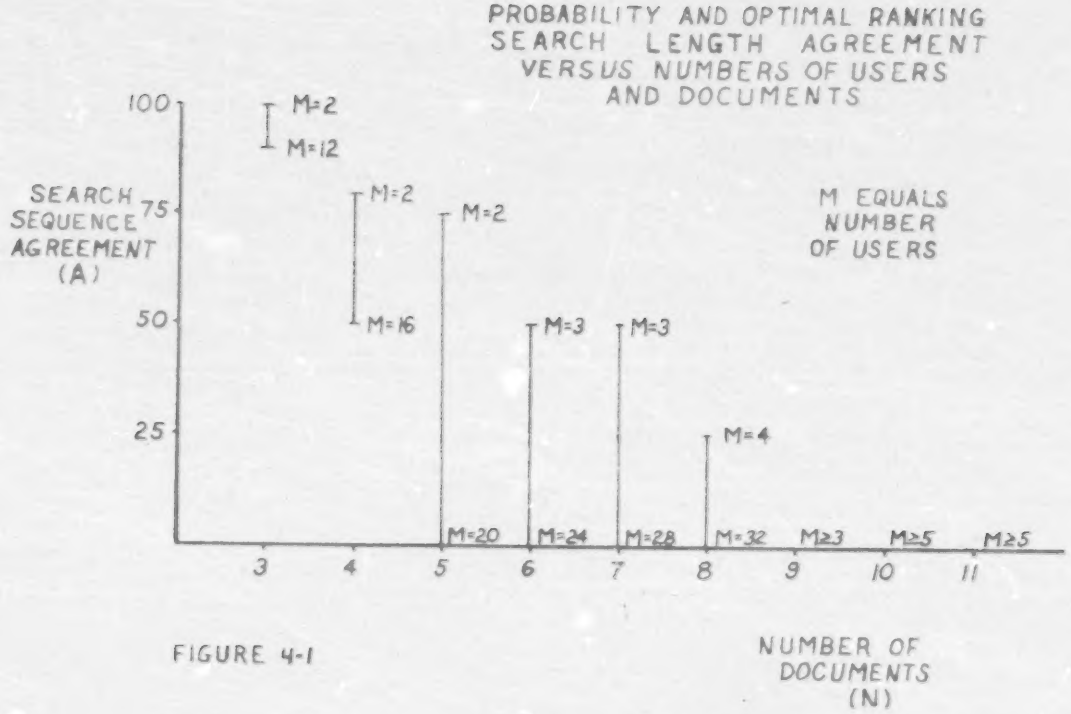

PROBABILITY AND OPTIMAL RANKING
SEARCH LENGTH AGREEMENT
VERSUS NUMBERS OF USERS
AND DOCUMENTS
SEARCH SEQUENCE AGREEMENT (A)
100
75
50
25
M=2
M=12
M=2
M=16
M=2
M=3
M=3
M=4
M=20
M=24
M=28
M=32
M≥3
M≥5
M≥5
3
4
5
6
7
8
9
10
11
M EQUALS NUMBER OF USERS
NUMBER OF DOCUMENTS (N)

FIGURE 4-1

probability ranking is not much different than the optimal technique. However, when the number of documents increases, the two methods disagree most of the time.

Probably of greater significance is the degree of degradation of search length due to probability ranking when compared with optimal ranking. For various numbers of users and each number of documents four* simulated matrices were randomly selected and reduced to a single data point by defining a measure of improvement, I, as:

$$I = \frac{1}{4M} \sum_{j=1}^{4} (\text{PRR Search Length})_j - (\text{ORR Search Length})_j \qquad (2)$$

For each value of N this measure yields the average number of nonrelevant documents each user does <u>not</u> encounter (due to the ORR improved ranking) while he is achieving his stop-search point.

Four models relating the average improvement, I, to numbers of documents, N, and numbers of users, M, were used as the basis for multiple regression analysis. These models were:

1) $I = \alpha_1 + \beta_1 N + \gamma_1 M$; 2) $I = \alpha_2 + \beta_2 (M/N)$;

3) $I = \exp\left[\alpha_3 + \beta_3 N + \gamma_3 M\right]$; 4) $I = \exp\left[\alpha_4 + \beta_4 (M/N)\right]$.

The variance in the dependent variable, I, accounted for by the independent variables, N and M (collectively), are 70%, 25%, 73%, and 30%, respective to the four models. The data are plotted in Figure (4-2) on semilog paper. The curves drawn represent the best fit to the data according to model #3, above, and for the values of M specified. The

---

*For N=11, because of high costs, only 2 matrices were generated for each value of M.

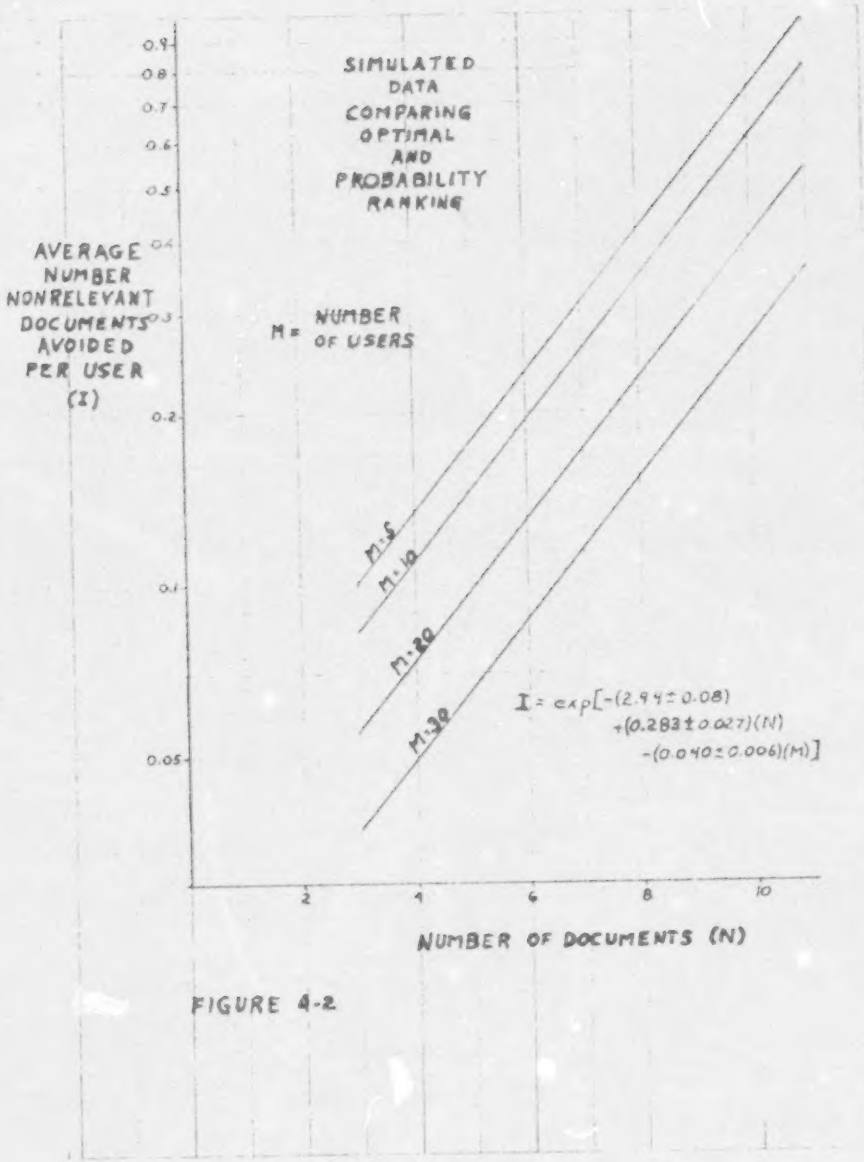

SIMULATED DATA COMPARING OPTIMAL AND PROBABILITY RANKING
AVERAGE NUMBER NONRELEVANT DOCUMENTS AVOIDED PER USER (I)
M = NUMBER OF USERS
M=5
M=10
M=20
M=30
I = exp[-(2.94±0.08) +(0.283±0.027)(N) -(0.040±0.006)(M)]
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.05
2
4
6
8
10
NUMBER OF DOCUMENTS (N)

FIGURE 4-2

error intervals on the regression coefficients are for one standard deviation.

Note the antagonism in the effect which M and N have upon I. Although the negative coefficient of M indicates that for extremely large values there will tend to be no improvement when comparing the ORR with the PRR, the fact that the regression coefficient is seven times greater for N suggests that the number of documents retrieved will dominate the expression. Consequently, as the number of documents within the retrieved set increases the patrons will encounter more nonrelevant ones under the PRR as compared with the ORR.

Figure (4-2) indicates that for the range of values represented the decrease in average search length per user under the ORR is not excessive, but if extrapolation to a value, for example of N=20 (while M = 10) is permissible, then each user on the average will search through ten less nonrelevant documents than he would under ranking by the PRR. If the definition of I used in Figure (4-2) is redefined according to:

$$PI = \frac{1}{4} \sum_{j=1}^{4} \frac{(\text{PRR Search Length})_j - (\text{ORR Search Length})_j}{(\text{PRR Search Length})_j} \times 100 \qquad (3)$$

then Figure (4-3) represents the percentage improvement in the users' average search lengths. For example, when M=N=10 (a square user-document matrix) the average improvement in the users' search lengths equals 26%. The fact that this percentage is equivalent to 6/10 nonrelevant document (from the regression equation of Figure (4-2), simply means that this search length under the PRR, as configured in the simulated data, is not very large. (For small values of N many were equal to zero.)

As a final observation drawn from Figures (4-1 and 4-2) one may

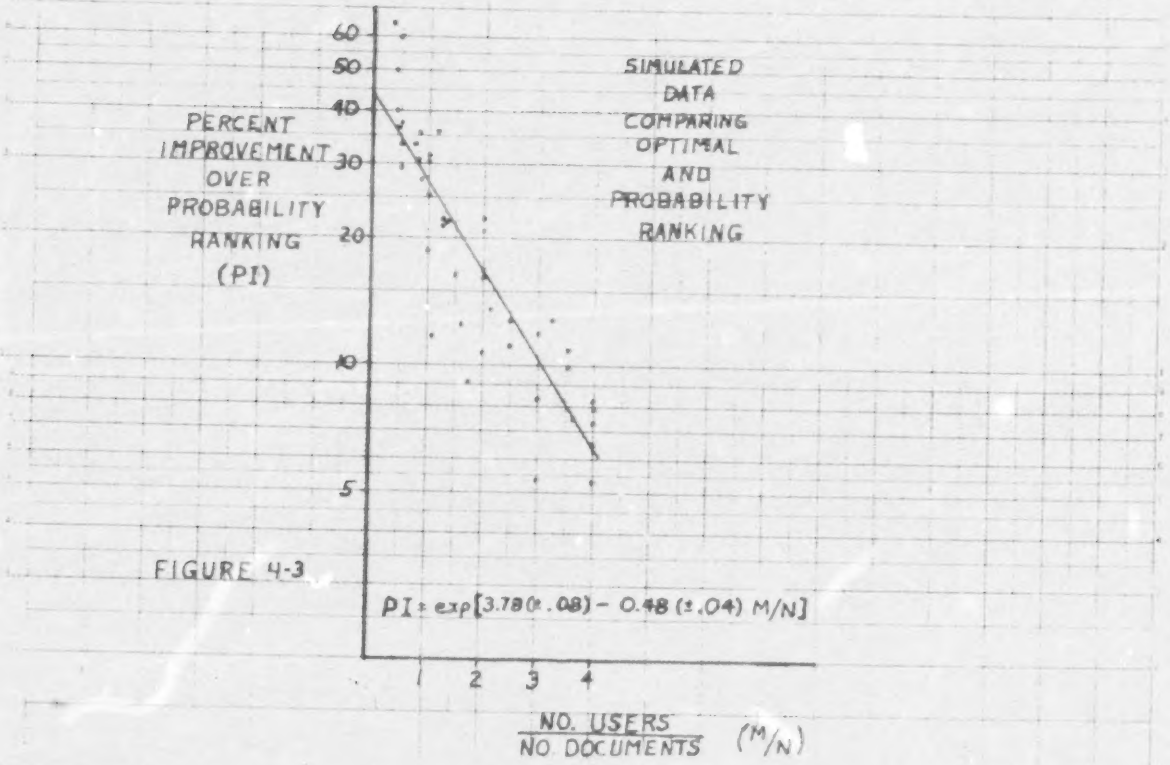

60
50
40
30
20
10
5
PERCENT IMPROVEMENT OVER PROBABILITY RANKING (PI)
SIMULATED DATA COMPARING OPTIMAL AND PROBABILITY RANKING
PI = exp[3.78(±.08) − 0.48 (±.04) M/N]
1
2
3
4
NO. USERS / NO. DOCUMENTS (M/N)

FIGURE 4-3

compare the effect displayed by the variable N with the results of Case III of chapter three. In the latter the limiting value of N=2 was shown to provide optimal ranking under the PRR. The figures just discussed also indicate that any advantage that the ORR has becomes negligible when N approaches two. So the empirical evidence generated in this chapter confirms one of the theoretical conditions for which the PRR is optimal.

Validating the Simulation

Certainly, a criticism of the simulation method used could be that evidence equating actual user judgments with the random patterns generated may not exist. The randomized, relevant-document generator makes possible all combinations of user relevance judgments, but to tie the model to reality, sampled data in the form of frequencies (some may be zero) need to be associated with each combination. Relevance patterns generated for users that frequently or infrequently occur, or who are nonexistant can then be controlled. Only by weighting the simulated matrices in the foregoing manner can one assure himself that the simulated patterns reflect a true cross section of the population. On the other hand, though, one is assured that the simulator does generate patterns of relevance judgments which actual users could produce since it can produce all possible patterns. Evidence relating the model of user relevance judgments to operational retrieval systems lies outside the domain of this study, and in the absence of this data the randomized procedure is the best* approximation

---

*The errors between the results represented by the simulation model and the results one would observe in real life are simply that the model generates relevance patterns that may not occur in real life, and it may not render sufficient weight to those patterns that do occur in real life.

Besides the question as to whether or not the model of user relevance judgments correctly reflects reality, one must also determine whether or not the data analysis procedure permits proper interpretation of the data generated* by the model. Multiple regression analysis provides several aids to help in this regard and these will now be discussed.

The multiple regression routine used was version V602 of the Statistical Package For the Social Sciences, SPSS. [44, pp. 320-67]. For Figure (4-2) the following results were observed. A weak, negative simple correlation existed between I and M, r= - 0.20; a strong positive correlation existed between I and N, r = + 0.69; and a fairly strong interdependence occurred between M and N, r = + 0.38. When M was entered as the first independent variable into the regression equation, only 4% of the variance in I was accounted for--a value that was not statistically significant. However, when N was entered as the first variable 48% of the variance in I could be explained.

$$(F = 39.1,\ df = 1, 42.\ P < .01)$$

When M was entered as the second variable, an additional 25% of the variance in I was accounted for.

$$(F_N = 106.5,\ df = 1, 41.\ P_N < .01;\ F_M = 38.5,\ df = 1,41.\ P_M < .01)$$

Collectively, M and N accounted for 73% of the variance in I and produced a statistically significant regression equation.

$$(F = 56.2,\ df = 2,42.\ P < .01)$$

The variance in PI that could be accounted for by M and N with regard to the four regression models assumed in the previous analysis was,

---

*Ordinarilly a simulator would produce data ranging over the entire gamut of possible values. Due to cost limitations on the simulator the retrieved numbers of documents range only from three to eleven.

respectively, 59%, 63%, 69%, and 75%. The fourth model was selected and the ratio of M and N was treated as one variable. In this case M tends to be the dominating variable with a strong, negative correlation between M and PI of -0.73, a very weak correlation between N and PI of +0.08, and a strong negative correlation between M/N and PI of -0.86. The interdependence between M and N is still the same with a correlation of +0.38. The regression equation resulting for PI and M/N is expressed graphically in Figure (4-3).

$(F = 124.1, df = 1,42. P < .01)$

When working with percentages, particularly those associated with PI, equation (3), a false illusion of improvement may appear--since the value of the denominator in equation (3) is changing. This effect may be illustrated by contemplating the actual amount of pizza three persons would obtain if each received 30 percent of a six, ten, and twelve inch pizza, respectively. In the case of PI often the values for the PRR search length would be zero or small when the values of N were small and those for M even smaller. Thus, in these instances, small improvements caused by the ORR would yield high percentage improvements when compared with the PRR. Aside from those instances in which N is small and $M/N<1$, Figure (4-3) presents a striking effect which the dimensions of the N by M array have upon percent average improvement. Any misunderstanding that this figure may convey is kept in check by Figure (4-2)--which indicates the actual amount of document(s) represented in the improvement.

In conclusion, legitimate disputations frequently do exist among patrons with similar requests because of differing perceived information needs. Even though these disputations may not be totally congruous with

those developed by the simulator, all that is required for the PRR's sub-optimal condition is, simply, that disputations in users' relevance judgments--akin to those described on page 43, chapter three--exist over more than two documents.

The multiple regression equations, such as those represented in Figure (4-2 and 4-3), may be used to either describe or predict. To justify their predictive validity, one must ensure first, that the simulated data is equivalent to randomly selected data from the user-document populations and secondly, that the data reduction procedure is statistically significant. Results from this chapter indicate that the second condition has been met. In the absence of validating data regarding the first point, we argue that any error in the regression equations can be attributed either to those patterns generated by the simulator for which no user would exist, or to those patterns for which users exist, but insufficient weighting has been attached to the relevance judgment pattern in question. Thus, we assert that the regression model associated with Figure (4-2) would be stable upon the entry of validating data and only the regression coefficients would change. This model will be used in chapter six to identify the cost/benefit "break even" point between the ORR and the PRR.

Having presented both analytic (chapter 3) and empirical evidence to the suboptimality of the PRR, we now turn our attention to the optimal ranking algorithm itself.

Chapter 5

THE OPTIMAL RANKING ALGORITHM

In this chapter the optimal ranking rule will be defined. Since the PRR often produces less than the best document ranking, for both theoretical and practical considerations the analysis of the ORR will be of interest. Consistent with the proof in Chapter Three and the simulated data in Chapter Four, user utility has been reduced to binary relevance judgments. Under this simplification the algorithm will be analyzed for efficiency. Although enormous improvement is achieved as compared to an exhaustive evaluation, the optimal algorithm is identified as requiring exponential time and space and thus remains in the class of computationally difficult problems.

Formulating the Problem

Numerous counter examples to the superiority of probability ranking were contrived. A few other counter examples were stumbled upon, but no efficient algorithm had been developed that would guarantee an optimal ranking for document retrieval systems. One could always use an exhaustive, brute force approach and evaluate the utility for all possible sequences of retrieved documents - selecting the sequence which minimizes the negative utility for all users. But this approach would be tiresome, inefficient, and frequently impossible. To illustrate the complexity of the problem, consider the following formulation:

$$C_{<i_N,\, i_{N-1},\ldots,\, i_1>,\, k} \begin{cases} =1, & \text{if the } k^{th} \text{ user judges the } i_N\text{th document to be not* relevant to his request after having retrieved the sequence of documents } <i_{N-1},\, i_{N-2},\ldots,i_1>. \\ =0, & \text{if the } i_N\text{th document is relevant to the } k^{th} \text{ user after having retrieved the sequence of documents } <i_{N-1},\, i_{N-2},\ldots,\, i_1>. \end{cases} \quad (1)$$

$$F_{<i_N,i_{N-1},\ldots,\, i_1>,\, k=} \begin{cases} 1, & \text{if the } k^{th} \text{ user's search has not been completed prior to document } i_N \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$D_{<i_{N-1},\, i_{N-2},\ldots,\, i_1>,\, i_N=} \begin{cases} 1, & \text{if all users are to evaluate document } i_N \text{ after receiving the sequence } <i_{N-1},\, i_{N-2},\ldots,\, i_1> \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Now for M users, the decision function for sequencing <u>two</u> documents would be determined by computing the cost of all paired sequences and selecting the minimum, that is

$$\min \left[\sum_{i_1=1}^{2} \sum_{i_2=1}^{2} \sum_{k=1}^{M} C_{<i_2,i_1>,\, k}\, F_{<i_2,i_1>,\, k}\, D_{<i_1>,\, i_2} \right] \quad i_2 \neq i_1$$

---

*Negative utility has (arbitrarily) been assigned a positive value. The sign convention selected was prompted through thinking of minimizing the "cost" (i.e., negative benefit) to the user.

For <u>three</u> documents the expression becomes

$$\min \left[\sum_{i_1=1}^{3} \sum_{i_2=1}^{3} \sum_{i_3=1}^{3} \sum_{k=1}^{M} C_{<i_3,i_2,i_1>,\,k} \; F_{<i_3,i_2,i_1>,\,k} \times D_{<i_2,i_1>,\,i_3} \right]$$

under the conditions $i_2 \neq i_1$

$i_3 \neq i_1, i_2$

And, finally, the generalized expression for N documents becomes:

$$\min \left[\sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \cdots \sum_{i_N=1}^{N} \sum_{k=1}^{M} C_{<i_N,i_{N-1},\ldots,i_1>,\,k} \times F_{<i_N,i_{N-1},\ldots,i_1>,\,k} \times D_{<i_{N-1},i_{N-2},\ldots,i_1>,\,i_N} \right]$$

under the conditions $i_2 \neq i_1$

$i_3 \neq i_2, i_1$

$\vdots$

$i_N \neq i_{N-1}, i_{N-2}, \ldots, i_1$ (4)

Observing the exclusions on each index $i_2, i_3, \ldots, i_N$ there are $[N(N-1)(N-2)\ldots(2)(1)]M = (N!)M$ sequences to be evaluated. Using Stirling's approximation, $M(N!) \approx M\left[\frac{N}{e}\right]^N \sqrt{2\pi N}$, which is of order $M(N)^{N+1/2}$. Thus, to evaluate all possible sequences would incur exponential costs, and such algorithms are said to be inefficient.[45, p. 364]. In contrast "efficient" algorithms require no more than polynomial time or space. (Polynomial functions are of the order $N^1+N^2+N^3+\ldots$, whereas

exponential functions are of the order $2^N, 3^N, 4^N, \ldots$, etc.)

Staging the Solution

Considerable economy can be achieved by "staging" a dynamic solution to the N factorial exhaustive approach. Staging permits one to create the optimal sequence of documents — one document at a time, by eliminating nonfeasible sequences at each stage on the basis of their intermediate cost (negative utility) functions. Specifically, a document is first selected to be the terminal one in a sequence. Since the cost for selecting a second document to be added to the sequence is known for all users (because of the relevance judgments contained in the user-document matrix and knowledge of each user's stop-search point), the running cost may be accumulated from one stage to the next. The cost functions at each stage for those sequences, which are merely permutations of each other, are compared and all but the one having the minimum cost are eliminated. The additivity feature of the cost function is essential in order that the decision consequences at each stage be explicit. [46, p. 332].

For example, assume that four documents (N=4) have been retrieved, and as portrayed in Figure 5-1, each is positioned at the first stage in a potential sequence. As the terminal (first stage) document in the sequence, each of the documents could be selected. Now at the second stage there are three (N-1) possible candidates to be prefixed to each of the four initial selections. We note that (at stage 2) the order 3-4 in sequence A is simply the inverse of 4-3 in sequence B. All decisions pertaining to this document pair will be the same at future stages--that is, two of the twenty-four possible sequences will be identical except one will end 3-4 and the other will terminate with 4-3. If we know the

value each user attributes to these two documents and what each user's stop-search point is, we may select the sequence having the least net cost--thus eliminating one of the possible twenty-four sequences at the second stage. In fact, there are ten additional paired sequences at this stage of which half may also be eliminated.

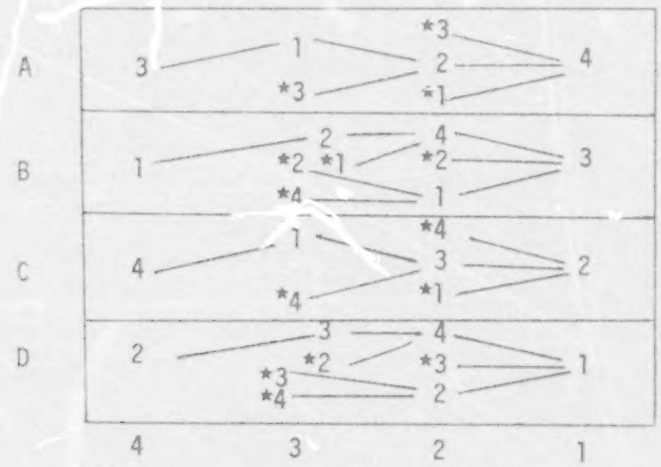


Figure 5-1

An Optimal Four-Document Sequence By the Backtracking Method

Note: The presence of an (*) indicates that the directed sequence has been terminated in favor of one of the other permutations of the same sequence. Thus, it is assumed that the cost of, for example, 2-3 is greater than the cost of 3-2 at stage 2, and the costs of 4-3-2 and 3-2-4 are each greater than the cost of 2-4-3 at stage 3.

At the third stage there exists two (N-2) possible candidates to be attached to each of the remaining $(N-2)\binom{N}{2} = 12$ pair sequences. Adding the "costs" of documents being attached at the third stage to the costs associated with the pairs surviving the second stage cut, results in a net cost for each sequence from its third stage document to its terminal

document. Following the procedure in the preceding paragraph, we note that (for stage 3) the order 1-2-4 in sequence A is simply a permutation of 2-4-1 and 4-2-1 in sequence D. Selecting the minimum cost triplet, only one of these three sequences needs be preserved for the last stage. And, in fact, two thirds of the triplet sequences which are presently existing at this stage will be terminated by similar analysis.

Finally, at the fourth stage there exists one (N-3) possible candidate to be attached to each of the remaining triplet sequences. Adding the cost of the fourth stage document being attached and that of the directed triplet preserved from the third stage, the optimal sequence may now be identified as the one having the minimum cost.

Thus the economy of the dynamic algorithm results from eliminating all but one of the permutations for each combination existing at any stage. Since each permutation has potential parenthood of numerous siblings at further stages, the savings in computation can be massive.

An Example Using the Staging Solution.

To illustrate, numerically, the previous example consider the user-document matrix of Figure 1-2 and reproduced in Figure 5-2. Table 5-1 represents the "cost" in going from the "next to the last" document to the "last" document. Implicit in this table is that the last document selected for the paired sequence will be the terminal document for those users who regard it to be relevant. Those users for whom the terminal document is not relevant have achieved their stop-search points earlier.

Users

Documents	1	2	3	4	5	6	7	8	9	Row
1	x	x	x	x	x					1
2						x	x	x	x	2
3	x			x			x			3
4		x			x			x		4
5			x			x			x	5

(Search length = 18)

DOCUMENT RANKING PRODUCED BY THE PRR

Figure 5-2

Document Sequence ↓	1 2	2 1	1 3	3 1	1 4	4 1	1 5	5 1	2 3	3 2	2 4	4 2	2 5	5 2	3 4	4 3	3 4	5 3	4 5	5 4
No. Nonrelevant Documents	4	5	1	3	1	3	2	4	2	3	2	3	1	2	3	3	3	3	3	3
Sequences Eliminated		*		*		*		*		*		*		*		* +		* +		* +

SECOND STAGE DECISION TABLE

Table 5-1

+ Since the "Cost" is equal for each of these inversions, the pairs eliminated were arbitrarily selected.

The sequences surviving the second stage and their "costs" have been transferred to Table 5-2. Each has been triplicated with one of the non-comon numbers prefixed. Two out of three sequences will be eliminated in this table. For example, 3-1-2 costs 10, 2-1-3 costs 6, and 1-2-3 costs 6. In this case 3-1-2 and 2-1-3 were eliminated. The remaining sequences will be transferred to Table 5-3, corresponding to the fourth stage, and those sequences surviving this stage will be transferred to the last stage depicted in Table 5-4.

Document Sequence →	No. Nonrelevant Documents	Sequences Eliminated
3-1-2	6+4	*
4-1-2	6+4	*
5-1-2	6+4	*
2-1-3	5+1	*
4-1-3	4+1	*
5-1-3	5+1	*
2-1-4	5+1	*
3-1-4	4+1	*
5-1-4	5+1	*
2-1-5	5+2	*
3-1-5	5+2	*
4-1-5	5+2	*
1-2-3	4+2	
4-2-3	5+2	*
5-2-3	4+2	*
1-2-4	4+2	
3-2-4	5+2	*
5-2-4	4+2	*
1-2-5	4+1	
3-2-5	4+1	
4-2-5	4+1	
1-3-4	2+3	
2-3-4	4+3	
5-3-4	6+3	*
1-3-5	3+3	
2-3-5	3+3	*
4-3-5	6+3	*
1-4-5	3+3	
2-4-5	3+3	*
3-4-5	6+3	

THIRD STAGE DECISION TABLE

Table 5-2

Document Sequence ↓	41235	51234	31245	51245	31245	41255	13255	43255	14255	34255	21345	51345	12345	52345	21355	41355	21455	31455	13455	23455
No. Nonrelevant Documents	6+6	6+6	6+6	6+6	6+5	6+5	4+5	6+5	4+5	6+5	5+5	6+5	4+7	6+7	5+6	6+6	5+6	6+6	4+9	5+9
Sequences Eliminated	*	*	*	*	*	*		*					*	*	*	*	*	*	*	*

FOURTH STAGE DECISION TABLE

Table 5-3

Document Sequence ↓	41325	31425	13425	52134	25134
No. Nonrelevant Documents	6+9	6+9	4+11	6+10	5+11
Sequences Eliminated (or Search Length)	15	15	15	*	*

FIFTH STAGE DECISION TABLE

Table 5-4

Any one of the three sequences in Table 5-4 is optimal since each has a search length of 15 documents. Sequence 4-1-3-2-5 is illustrated in Figure 5-3. Note that sequence 1-3-4-2-5 is also optimal.

		1	2	3	4	5	6	7	8	9	
	4		x			x			x		1
	1	x	x	x	x	x					2
Documents	3	x			x			x			3 Row
	2						x	x	x	x	4
	5			x			x			x	5

DOCUMENT RANKING PRODUCED BY THE ORR

Figure 5-3

Emphasizing the complexity of the optimizing algorithm we note that two pair sequences of the five document sequence in Figure 5-3, namely 4-1 and 3-2 were eliminated at the second stage in Table 5-1 because each was suboptimal at that stage. (Also, sequence 4-1-3 was eliminated at the third stage.) However, at the fifth and third stages, respectively, each pair sequence is now part of an optimal five document sequence. These results are at variance with those of Baker. [33, p. 207].

In general, if we have a value to the cost function for the first ℓ elements of a partial sequence, $C_{<i_1,i_2,\ldots,i_\ell>}$, and the remaining cost, $C_{<i_{\ell+1},i_{\ell+2},\ldots i_N>}$ has not yet been determined, then $C_{<i_1,i_2,\ldots,i_\ell>}$ must have the property that if the cost of any permutation of the partial sequence $<i_1,i_2,\ldots,i_\ell>$ violates the condition of

$\min \left[C_{<i_1,i_2,\ldots,i_\ell>}\right]$ then any completion of this partial sequence will also violate the end result of $\min \left[C_{<i_1,i_2,\ldots,i_N>}\right]$. Whitehead refers to this method as "backtracking". He further appropriately warns that the technique used to identify the partial sequences which are permutations of a common combination will directly determine the efficiency of the staging algorithm. [47, p. 69].

The Optimal Algorithm

1) Let $\{X_{ij}\}$ be an NXM document-user matrix where

$$X_{ij} = \begin{cases} 1, & \text{if the } i^{th} \text{ document is relevant to the } j^{th} \text{ user} \\ 0, & \text{if the } i^{th} \text{ document is not relevant to the } j^{th} \text{ user} \end{cases}$$

N = Number of documents

M = Number of users

2) Consider stage k=2. Calculate all possible k-tuples of the N numbers taken 2 at a time, $\binom{N}{k}$.

3) Calculate the search length for each user at each stage, k, in going from the N-k+1 document to the N^{th} document.

$$SL = C_{<i_N,i_{N-1},\ldots,i_{N-k+1}>} = C_{<i_N,i_{N-1},\ldots,i_{N-k+2}>} + C_{<i_{N-k+2},i_{N-k+1}>}$$

$$= \sum_{j=1}^{M} \sum_{i=N}^{i=N-k+1} \left[\bigcup_{i=N}^{i=N-k+2} X_{ij}\right](1-X_{ij}) + \sum_{j=1}^{M} \left[\bigcup_{i=N}^{i=N-k+1} X_{ij}\right](1-X_{N-k+1,j}) \quad (5)$$

NOTE: (i) The range on i in both the sum and union expressions is the reverse of convention; i=N,N-1,...,2,1.
(ii) The first expression on the right of equation (5) is simply the value of equation (5) at the previous stage.
(iii) If the union expression in equation (5) is preserved for each user at each stage, it may be combined with the X_{ij} of the next added document to form the union of the second expression on the right of equation (5).

4) For each possible combination of k documents, eliminate all but the permutation with the smallest search length value from further consideration.

   NOTE: Preserve this value, the corresponding document sequence, and the accumulative union of the X_{ij} for each user and pass these to the next stage.

5) If k=N, output the optimal document sequence(s) as those surviving step (4) and stop. If k < N, go to step (6).

6) For each k-tuple surviving step (4), form a series of new (k+1)-tuples by adding a document number not in the original k-tuple. This number will be placed at the beginning of the new tuple. Do this for all possible numbers in the range [1, 2,...,N] not in the original tuple.

7) Set k=k+1. Go to step (3).

Conveniently, the backtracking algorithm as described above permits one to maintain the F_{ik} (i.e., the "stop search values"), simply, by performing a logical "OR" on the X_{ij} for each user--beginning with the terminal document and continuing according to the particular sequence. Not having the F_{ik} is equivalent to forcing all users to read all documents.

Analysis of the Optimal Algorithm

Earlier it was shown that the exhaustive approach would entail evaluating N! different sequences. But within each sequence, and at each stage there are numerous cost functions to be compared. To analyze the efficiency of the solution algorithm, it will be instructive to estimate the number of cost comparisons necessary.

Using the exhaustive approach - that is, computing a cost function

at all stages of the N! sequences requires*

$$M\left[N(N-1) + N(N-1)(N-2) + \ldots + N(N-1)(N-2)\ldots(3)(2) + N!\right]$$

or,

$$MN!\left[\frac{1}{(N-2)!} + \frac{1}{(N-3)!} + \ldots + \frac{1}{(1)!} + \frac{1}{(0)!}\right]$$

which is equivalent to

$$M \sum_{k=0}^{N-2} \frac{N!}{k!} \quad \text{cost functions.}$$

Referring back to the example of Figure (5-1), for N=4 there are

$$M \sum_{k=0}^{2} \left(\frac{24}{k!}\right) = 24\left[\frac{1}{1} + \frac{1}{1} + \frac{1}{2}\right] = M(60) \text{ cost functions}$$

within the twenty-four sequences that would be computed under the exhaustive approach.

In contrast the number of cost functions required by the ORR is (derivation is in Appendix F):

$$\text{total cost functions} = MN\left[2^{N-1}-1\right] \qquad (6)$$

Referring to Figure (5-1), again, equation (6) indicates that the number of cost functions computed under the optimal ranking algorithm would be $M4\left[2^{3-1}\right] = M(28)$. In this example the number of cost comparisons is less than half the amount required by the exhaustive algorithm.

Equation (6), representing the total number of cost functions evaluated by the algorithm, agrees precisely with observed results. For a given value of N it represents a much lower exponential than the number

---

*The arbitrary number of users in the development will simply be represented by M.

of cost comparisons obtained under the exhaustive procedure, but unfortunately, is still exponential. The exponential nature of the algorithm is portrayed in Figure (5-4). Each of the components (i.e., time for optimally ranking a square matrix and the minimum number of memory locations required) is plotted as a function of number of documents. (Note that the number of users are also increasing since the matrix is square.)

The graph relating to memory locations represents only a proportional, not actual, amount - since, for each sequence preserved at each stage, both its accumulative cost and a "flag vector" (i.e., $F_{i,k}$) must also be stored. The memory requirements for N=11 are only 17k, but climbing fast. Note that the slope of the time curve is steeper (almost vertical) than that for memory when N>10. Both curves signal that the practical limitations for the programmed version of the ORR are near.

The time required to determine the optimal sequencing for 11 documents (with 11 users) is running 9 3/4 minutes at a cost of around $50. Originally, it was hoped that the algorithm could accommodate N = 17 in order to evaluate a reference source problem. But even if the programmed version of the algorithm could be made more efficient, it is doubtful that six additional rows and columns on the document-user matrix would be feasible. The optimal ranking program may be found in the Appendix A.

Combinatoric problems, which the optimal ranking rule is a type, embrace such fields as scheduling, network optimization, and graph theory. These problems, requiring the scheduling of mutually exclusive events, need exponential time or space for solution.

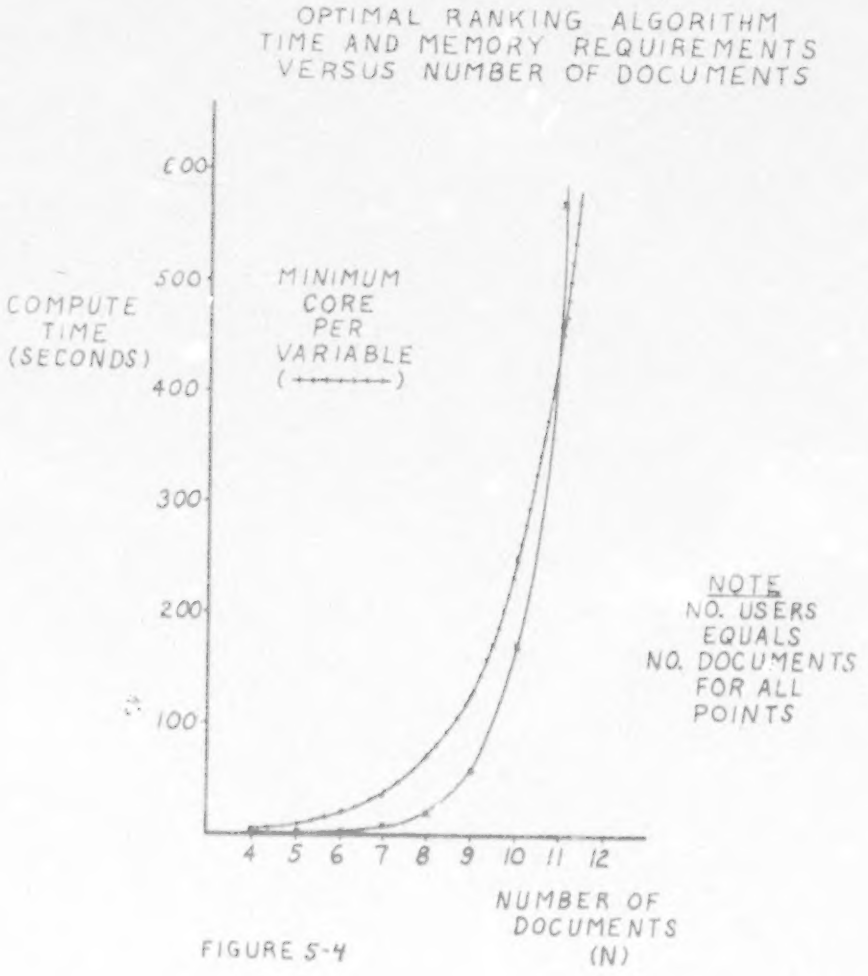

OPTIMAL RANKING ALGORITHM
TIME AND MEMORY REQUIREMENTS
VERSUS NUMBER OF DOCUMENTS
COMPUTE TIME (SECONDS)
600
500
400
300
200
100
MINIMUM CORE PER VARIABLE
4 5 6 7 8 9 10 11 12
NUMBER OF DOCUMENTS (N)
NOTE
NO. USERS EQUALS NO. DOCUMENTS FOR ALL POINTS

FIGURE 5-4

The document ranking problem presented here is similar to a version of the traveling salesman problem. It will be helpful to describe the latter since if an equivalency does exist between the two, certain theoretical bounds on the efficiency of the optimal ranking rule may exist. Briefly stated the traveling salesman problem is: Given the set of costs incurred while traveling between each of N cities, what is the sequence of cities which minimizes the total cost to tour each city once? Extending the problem to include k different salesman where each has different "costs" associated with the cities (which is analogous to the document users having different information needs), the similarity between obtaining the optimal document ranking and determining the minimum cost tour seems evident. The difference still remaining between the two models however, centers in the stop search points (i.e., the F_{ik}) associated with the ranking problem. Here each patron discards the remaining nonrelevant, but unread documents after reaching his stop search point. The documents which are members of the discarded group are determined by selecting the sequence for which the number of unread nonrelevant documents is maximum. Were the stop search points to be removed, the performance of the retrieval system would be the same for all sequences since all patrons would read all retrieved documents. Consequently, the "cost" associated with a specific nonrelevant document will be variable, since it depends on whether or not its position in the patron's search sequence lies before or after his stop search point. On the other hand, the cost associated with each city on the salesman's tour is fixed before the solution of the problem and in fact is required in order to determine the minimum cost tour. The dissimilarity between the two models, then, is in the uncertainty of the cost of a nonrelevant document before the

optimal sequence is determined, whereas in the tour problem all costs associated with cities are known in advance. In conclusion, the optimal (guaranteed) solution to both problems requires that the cost for each possible (in principle*) permuted sequence be computed so that the minimum cost sequence(s) may be selected. The nature of these solution procedures is N! which requires exponential time or space.

NP Complete Problems

While it has not been demonstrated here that another more efficient solution for the document ranking problem does not exist, the backtracking solution technique seems to be the only method that will accommodate the uncertainty in cost of the nonrelevant documents and at the same time guarantee an optimal solution. If we assume that the backtracking solution is the most efficient algorithm for optimal document sequencing, then the assertion follows that the problem requires exponential time or space for solution. If this is true then limitations exist on one's ability to optimally sequence large numbers of documents--even with the use of sophisticated computers.

Although we have identified it to be slightly different, the traveling salesman problem is already associated with a class of computationally difficult problems. This class of problems is nondeterministic polynomial complete, or NP-complete. Karp has shown that this class which includes the traveling salesman problem, is equivalent ("complete") in that either all of the problems within the class have polynomial bounded

---

*The costs for the majority of sequences need not be computed since (as earlier shown) numberous unfeasible sequences may be recognized and eliminated during the solution procedure.

solution algorithms (which seems most unlikely, according to him) or none of them do. [48, pp. 93-97]. See, also [49]. As yet, no one has proved that efficient (polynomial bounded) algorithms exist for solving NP-complete problems. [50].

Recently, Yu also showed that the class of computationally difficult NP-complete problems occurs in another phase of document retrieval systems. Using a mathematical model of the semantic relations between index terms he shows that improvement in retrieval system performance can be unexpectedly large. However, the solution algorithm is NP-complete which requires an enormous amount of computing time.[51, pp. 185-6].

If future research demonstrates that the optimal document ranking algorithm is NP-complete, then it would be advisable to focus on efficient heuristic ranking procedures, such as those described in chapter one, rather than costly algorithms.

In the next chapter we will look at the cost/benefit trade off of the PRR as compared with the ORR and field several procedural and conceptual objections.

Chapter 6

THE OPTIMAL RANKING RULE--REVISITED

Utilizing the data generated in chapter four which compared the PRR and the ORR search lengths and the efficiency of the ORR as determined in chapter five, a performance measure relating the two will be developed in this chapter. A cost/benefit trade off between the two ranking rules will follow.

A really comprehensive evaluation of the PRR would also include comparisons with potentially feasible alternatives such as the ranking heuristics described in chapter one. This is proposed as later follow-up research. For the present, the evaluation of the PRR will be limited to comparing its efficiency against the ORR, which is used as a standard of optimal efficiency even though the heuristic approaches are likely to provide cost/benefit ratios that are preferable to either the PRR or the ORR. Concluding this chapter will be responses to several objections pertaining to the ORR.

Cost/Benefit Analysis

The analyses in chapter five determined the number of cost computations necessary under the ORR to be $MN(2^{N-1}-1)$. This function is proportional to the ORR's overhead in computing the optimal document sequence. With regard to the PRR, its overhead is proportional to NM (i.e., adding up the number of relevance "votes" for the N documents and M users), plus whatever cost is incurred in sorting the N documents by

relevance "votes" into decreasing order. Assuming that an efficient sorting routine is used, the combined overhead for the PRR is $N(M+\log_2 N)$. (The number of comparisons necessary under a heapsort routine are proportional to $N \log_2 N$. [52]. The difference in overhead between the two ranking procedures, then, is proportional to $NM(2^{N-1}-1) - N(M+\log_2 N)$. This latter expression is the excess number of cost comparisons required by the ORR as compared with the PRR, and may be regarded as their differential work function, W.

The differential benefit, on the other hand, in using the ORR in comparison with the PRR may be approximated by the regression equation of Figure (4-2),

$$I = [\exp -2.94 - 0.04\ (M) + 0.283\ (N)]$$

The latter expression predicts (restricted by the assumptions stated in chapter four) the number of nonrelevant documents each user will avoid on the average due to the ORR as compared with the PRR.

Combining the expressions representing differential benefit and differential work, a performance measure, P, is defined as

$$P = cI - \propto w \qquad (1)$$

where c and $\propto$ represent constants that transform "nonrelevant documents avoided" and "number of cost computations" into units of utility to the user. The negation sign preceding $\propto w$ indicates that this factor contributes a loss to the users utility function.

Expressing equation (1) in terms of N, M, c, and $\propto$ we have

$$P = c\{\exp[-2.94 - 0.04\ (M) + 0.283\ (N)]\} - \propto\{NM[2^{N-1}-1] - N[M+\log_2(N)]\} \qquad (2)$$

Equation (2) was programmed with varying values of N, M, and $c/\propto$ in order to identify the break-even points between the PRR and ORR. These points occur when P = 0. This data is plotted in Figure (6-1).

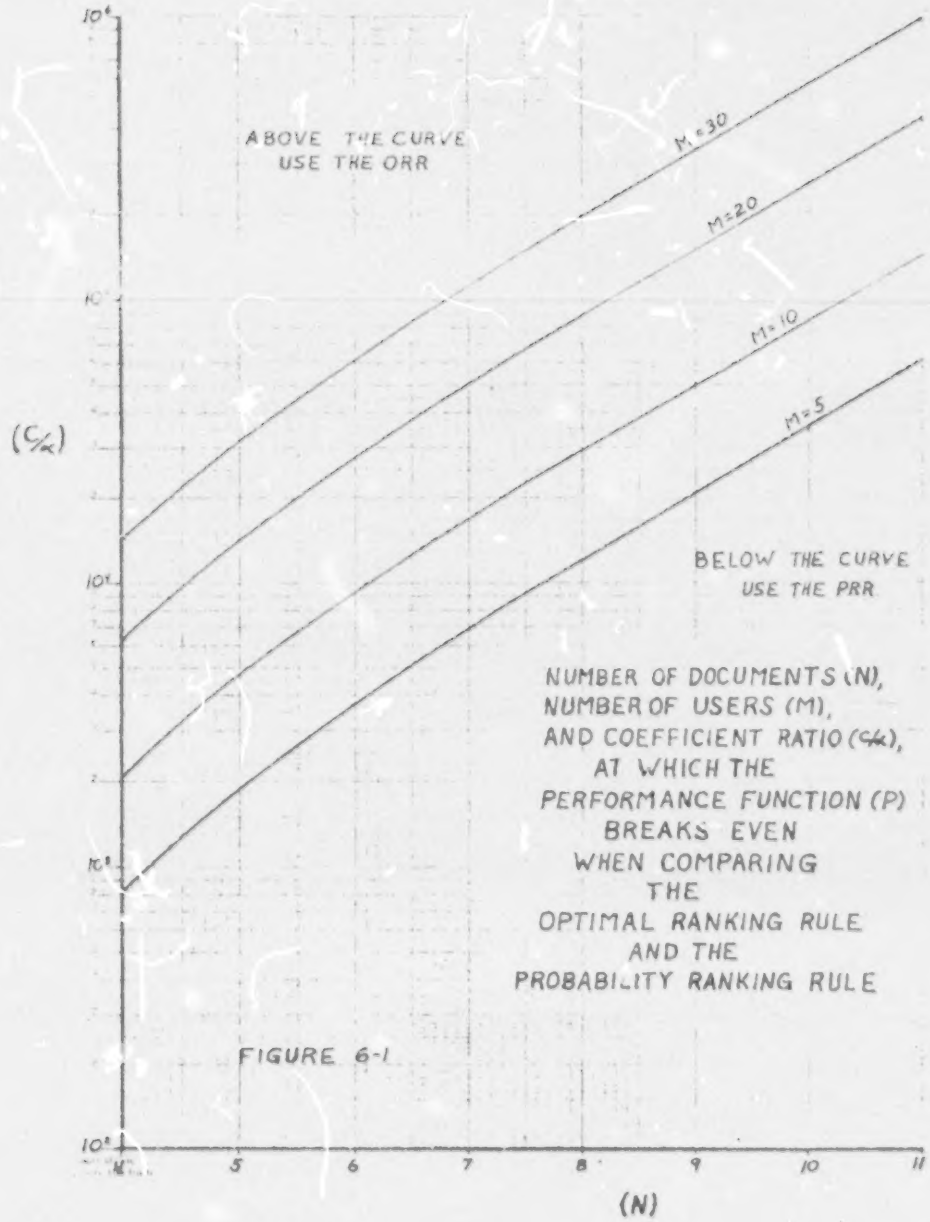

ABOVE THE CURVE
USE THE ORR
M=30
M=20
M=10
M=5
BELOW THE CURVE
USE THE PRR
NUMBER OF DOCUMENTS (N),
NUMBER OF USERS (M),
AND COEFFICIENT RATIO (C/c),
AT WHICH THE
PERFORMANCE FUNCTION (P)
BREAKS EVEN
WHEN COMPARING
THE
OPTIMAL RANKING RULE
AND THE
PROBABILITY RANKING RULE
(C/c)
(N)
5
6
7
8
9
10
11

FIGURE 6-1

Several important conclusions may be drawn from this figure:

1) Given the curve for M = constant, if the values of c/α and N (number of documents retrieved) intersect above the curve then use the ORR. On the other hand if the values of c/α and N intersect below the curve then use the PRR.
2) The value of c/α must be high when the number of documents increases in order for the ORR to be cost effective. No benefit exists from the ORR when $c/\alpha < 10^3$.

As discussed in chapter four the values of the regression coefficients in equation (2) are tentative. And, of course, as these vary so also will the curves in Figure (6-1)--but the procedure for analysis and decision would remain the same.

Were it not for the differential values between the coefficients c and α, there would be no break even point between the ORR and the PRR. A closer look at the benefit function, I, and the work function, W, reveals that a potentially mammoth difference in scale exists between the two. Where the average number of nonrelevant documents avoided per user in relation to ten documents retrieved differs by a factor on the order of ten, the number of cost computations in relation to the same ten documents retrieved differ by a factor on the order of 10^3. These order of magnitude estimates indicate that if the benefit gained by avoiding nonrelevant documents and work expended while determining cost functions are to be of the same order of magnitude, then α must be no greater than 10^{-3}. And further, with $\alpha = 10^{-3}$ (and c=1) Figure 6-1 indicates that one should always select the probability ranking rule in favor of the other ranking procedure.

While the previous decision may be valid, it assumes that benefit

and work are both ratio measures rather than interval--which may be questionable in this instance. The ORR requires exponential time and space to compute, and as such it is very unlikely that the user will be determining its cost computations himself. Relegating this effort to a computer seems to bring into dispute the equivalency of the two performance factors. On the one hand the patron is accorded the convenience of avoiding nonrelevant documents, while on the other hand a computer is grinding out cost computations for the staging algorithm. So with a quasiargument the assertion is made that, in this instance, benefit and work are not on an equivalent plane and α should be less than 10^{-3}.

What is the value of α? Conveniently, the negative utility represented by $-\alpha W$, the difference in the number of cost computations between the two ranking rules, could be replaced by CPU cost or time. Either of these computer charges would represent an objective estimate of the work function. But now we must raise similar questions about the value of c. How much in equivalent units of measurement (dollar cost or time) is "avoiding a nonrelevant document" worth to a user? With the current interest in online bibliographic retrieval, the question just raised is no stranger. Although some excellent research on estimating user utility is appearing in the literature, these procedures lie outside the scope of the research being reported here. (For example, see [39].)

Having identified the conditions under which one would use the two ranking rules, our attention is now turned to two more discussions pertaining to the availability of data--essential for the optimal ranking rule.

Objection #7

Sampling techniques have been suggested as an approach for obtaining data needed to characterize the user population. Presently, statistical theory is well developed for inferring, for example, that the sample mean is the best estimate of the population mean--for interval statistics. But at an ordinal level, sampling theory cannot guarantee that the rank ordering resulting from a sample is the best estimate of the rank ordering for the population. Since the optimal ranking rule requires rank orderings from all subgroups of the user population (user-types) so that an optimal ranking ordering (i.e., search strategy) may be constructed for the entire population, how may the subgroup rankings be obtained, short of restricting the subgroup population to one member? Reducing the membership for each subgroup to one is tantamount to increasing the number of subgroups to the size of the entire population, implying the necessity for a total population count rather than a sample.

Response

This objection stems from misunderstanding the nature of the source data used by the optimal ranking rule. A characteristic of ordinal data is that only an inequality relation holds between the data elements. It is known that a data element is "greater than" or "less than" another, but one is unable to quantify the magnitude of this difference. For example, General Motors Corporation may infer that a Buick is more prestigious than a Chevrolet, and that a Cadillac is more prestigious than a Buick, but without the use of an interval measure, such as cost, the consumer is unable to determine the magnitude of difference in prestige between a Cadillac and a Buick and between a Buick and a Chevrolet. To further complicate matters the Environmental

Protection Agency may equate prestige with driving an automobile which has low exhaust pollutants or low fuel consumption. In consequence, a different rank ordering of the vehicles may result. Constructing a single rank ordering from a group of sample rank orderings usually involves counting the number of first place positions, second place positions, etc., each vehicle has and then ranking them accordingly. In doing so, however, the ordinal data has been evaluated to either an interval or ratio level on two points. First, assigning the vehicles to first, second, and third place quantifies their relative differences onto a numeric interval scale, and secondly, adding up "the number of first place positions," etc., results in normalizing the value judgments of GMC and EPA onto a ratio scale by giving them equal weight. So, indeed, different rank orderings of data elements which must be retained in ordinal form have no basis for being averaged into a single rank ordering.

However, even though the optimal search strategy is contingent upon a rank ordering that is characteristic of the entire population, the utility estimates used to obtain the subgroup rank orderings are not ordinal, but interval data. Thus, we are not dealing with only an inequality relationship between documents as we were with automobiles, but each document is represented as having a numerical positive or negative benefit to the user.

Only partially resolved, though, is the problem of combining document utility estimates from different users. Lacking an absolute base (ratio statistic) one patron may use inflated utility estimates as compared with another. One way to field this problem would be to assume that another patron may assign "deflated" estimates so that on the

average the two will equalize. Or, perhaps, another approach would be to actually charge the patrons the value (or a portion of it) which they assign to the documents and let the economics of the market place control pricing. (Admittedly, there exist problems with the latter approach if one patron must rely totally on personal resources while another is subsidized by an employer.)

An appropriate sampling scheme would be to randomly sample document utility estimates from the set of patrons having similar requests (and retrieved output), sort these into homogeneous subgroups, and weight each subgroup by its frequency.

Objection #8

The Optimal Ranking Rule described in the Appendix and used in the simulation experiment of Chapter four is based upon binary relevance judgments rather than utility estimates. Although portrayed differently in Chapter five, relevance assignments to a document are not (either traditionally or in the simulation experiment of chapter four) affected by the material previously encountered by the patron. Thus, the optimal search strategies have been determined on the basis of document relevance judgments made in isolation from the influence of other documents. Most everyone would agree that the perceived value of documents read in some sequence is affected by those already perused.

Response

A totally satisfactory response to this objection, avoiding an N factorial solution, is not possible. Initially, it can be argued that relevance judgments, in general, should be replaced by utility estimates. In doing so the patron is <u>not</u> advised to isolate the influence that

previously read material has had upon him, but rather to assign the interval measure to the document at hand on the basis (for example) of how much he would be willing to pay for it. This estimate may have an absolute base, a differential (or incremental) base--that is, only the value that a document has added to the user's knowledge would be estimated, or a combination of both. The algorithm used in the simulation assumes an absolute value for each document whereas the formal development at the first of the previous chapter is not so restricted. In fairness to this "absolute" approach it is not absurd when one understands the goal of the retrieval system. Where we were concerned with document utility, presumably, the worth of the retrieved information was measured by a timely answer, lead, or set of references to aid the patron. The patron was viewed as having a "gap" in his mental map - requiring heretofore unobtained information to solve his problem.

Under the simulation model restrictions of Chapter four, however, a document tends to have a more absolute relationship with the request--one that may be less dependent upon the urgency of the patron's specific need. For example, although a document may have been already read by the patron, it could still have historical value to him while at the same time having little utility towards extending his knowledge or rendering additional problem solving capability.

Reducing the generalized concept of utility to a binary choice simply made the illustrations of chapter two more visible, albeit, the evaluation of the retrieval system may have appeared artificial.

A problem, however, arises in attempting to implement the unrestricted version of the optimal ranking rule. The general version permits the patron to assign a utility estimate to a document which

reflects only the additional benefit which he receives. Consequently, the worth of a document to a patron can vary not only by its position within a sequence, but also by the specific documents preceding the document in question and possibly, the order in which the preceding documents have been read. Thus, in principle, to apply the unrestricted version of the optimal ranking rule one would need to have patron utility assignments for all permutations of document orderings--which is the problem we are trying to avoid!

On the other hand, though, more frustration might be experienced by the patron if he were forced to assign only incremental (rather than absolute) benefits to documents. Mentally keeping tract of value already achieved as opposed to "new" value while reading a document would introduced another form of torture. While not directly advising the patron to make document utility judgments independent of material previously read, simply ask the patron to estimate the worth of the documents. Though not totally untainted from incremental effects, these values will likely be good approximations to absolute values as perceived by the user.

Having completed the comparative analysis between the ORR and the PRR, in the next section we will draw some conclusions as to the implications which the research performed here has upon information retrieval systems.

Chapter 7

CONCLUSION

Webster's dictionary defines optimal as the condition of being "most desirable or satisfactory." The approach taken in this report has been to identify a ranking criterion which is correct in principle even though it may be difficult or impossible to apply in practice. The rationale for this approach is that compromise, although frequently essential, is only meaningful if the consequence of alternatives is clear. In particular the cost of achieving optimal document ranking may exceed the benefit of obtaining the condition, but at least a bench mark exists from which to measure future retrieval effectiveness.

In this chapter the conditions for optimal and suboptimal performance of the PRR and estimates of the degradation to be expected when the PRR is suboptimal will be restated. The exponential nature of certain ranking algorithms will also be summarized. In addition, several alternatives to the optimal ranking algorithm developed in this report will be identified. Finally, a note about the importance of further research in this area will conclude this study.

Suboptimal Performance of the Probability Ranking Rule

Counter examples to the optimality of the simple probability ranking rule (PRR) were identified in chapter two. Recognizing that the counter examples may simply be anomalies which are atypical of normal performance, the conditions for suboptimal performance of the PRR were explored in

chapter three. There it was determined that the PRR principle will yield optimal rankings:

a) if all patrons will be satisfied with one and only one relevant document, or

b) if only two documents are retrieved. (The patrons may regard either document as being relevant or both documents as being relevant.)

Given that the retrieval system would yield more than two documents and that its patrons may require more than one relevant document to complete their searches, two questions were next posed: (1) how frequently do the suboptimal conditions occur and (2) how severe is the degradation of the performance of the PRR principle? A simulation study was reported in chapter four, which generated data in response to these questions.

Responding to the first question, the study found that less than half (41%) of the simulated document rankings produced by the PRR were optimal. Confirming findings from chapter three, the simulation results indicated that when the number of documents retrieved is very small, the results of the probability and optimal ranking rules are similar. However, when the number of documents increases, the PRR performs less than optimally most of the time.

Regarding the second question, the magnitude of improvement of the optimal ranking rule (ORR) with respect to the PRR is dependent upon both numbers of users and numbers of documents. Regression analysis of the simulated data indicated that an antagonistic effect upon ranking existed between users and documents. As the numbers of users increase the PRR approaches optimality, while as the numbers of documents increase the PRR's performance degrades. However, the number of documents clearly dominate the regression expression since its coefficient is seven times

that for the number of users. Results from the simulation indicate that for an equal number of users and documents (i.e., a square matrix), the ORR would on the average require 27% fewer nonrelevant documents to be searched. Quantitatively, at N=11 this percentage is equivalent to three fourths of a nonrelevant document per user. At N=20 and an identical number of users, there would be on the average six additional nonrelevant documents withheld from each user. Thus the ORR has significant potential for registering improved retrieval effectiveness in comparison with the PRR.

A closer look at the PRR suggested that the reason why it produced suboptimal search strategies is that it is indifferent as to whether or not a given document in a sequence will be terminal or nonterminal for a given user. If a document would be ranked highly by most of the users, but is nonterminal for them, it may be moved towards the end of the search sequence without penalty to them. Determining a document's optimal position within a sequence cannot be done solely on the basis of the number of users who regard it to be relevant. The document ranking example portrayed in chapter five (pages 66-70), illustrated clearly the permutation problem that occurs while sequencing documents. There, two paired sequences and one triplet sequence had been eliminated because they were suboptimal at one stage only to find that they became part of the optimal solution when considered at another stage in the solution sequence. In a retrospective system such as the one assumed in this report, an optimal sequence cannot be guaranteed unless the solution algorithm is sensitive to each user's stop-search point.

Neither the ORR nor the PRR is restricted to the binary relevance assignments used in the model depicted. Both ranking criteria are

adaptable to an interval utility estimate. However, summarizing individual utilities into a single statistic masks user conflict. For example, no difference would be perceived in a document having zero utility to two users when compared with one document having ten positive units of utility to one user and the same document having ten negative units of utility to another user. It is true that the net worth of both documents is the same, but the PRR is not sensitive to users' stop-search points and thus cannot exploit any benefit due to interaction with those stop-search points. Because of the summarizing effect of the PRR, documents for which conflict in user utility assessments exists would be consistently ranked low, whereas under the ORR they could be assigned virtually any position within the sequence. When satisfaction or dissatisfaction of several users is summarized into one statistic, a loss of information occurs causing greater uncertainty.

Computational Constraints on Optimal Ranking Procedures

Extending the simple PRR to an optimal strategy which minimized expected loss to the users (chapter two), required that N factorial sequences be computed ($N! \simeq N^N$). Doing so is computationally difficult for larger numbers of documents. In attempting to achieve the optimal document sequence for the more general case under which different users may have different stop-search points, the total number of cost functions to be computed under the N factorial, brute-force evaluation is

$MN! \sum_{k=0}^{N-2} \frac{1}{k!}$. Even for relatively small values of N, this expression approaches $M(N!)(e)$, where e is the base for the natural logarithm. For N=11 there are M(108,505,100) cost functions. On the other hand, the number of cost functions computed by the ORR would be

$MN[2^{N-1}-1] = M\left\{11[2^{10}-1]\right\} = M(11,253)$. (Although a dramatic difference exists between the number of cost functions computed by the two rules, there is considerable effort involved in the latter to pass the cost functions and partial sequences surviving from one stage to the next.) The latter expression is still exponential, and for large values of N the effort required to optimally rank the documents still becomes unwieldy.

Although representing different orders of magnitude, the brute-force approach (N!), the optimal strategy (OPS) for minimizing expected loss, the optimal ranking rule (ORR), and the utility--theoretic approaches cited from the literature in chapter one, each of these procedures belongs to a class of computationally difficult problems requiring exponential time and space for solution. These classes of combinatoric and sequential type problems encountered in information retrieval are more extensive than have been heretofore realized. However, Robert Floyd struck an encouraging note at the Symposium for the Complexity of Computer Computations by stating that even though these problems are sticky, at least we know they exist. We are better off now than we were a few years ago because now we know that we have a problem![53].

A Compromise

The flaws in probability ranking have been portrayed, and yet the cure (i.e., the ORR) appears to have its own problems. Several alternatives for avoiding the computational burden of the ORR while maintaining a near optimal solution are available. These will still accommodate a retrospective retrieval environment where individual differences among user document utilities have been preserved.

One alternative is to identify the conditions for which the PRR is optimal, or nearly so, and then attempt to control the numbers of users and documents for each retrieved set within these conditions. For example, earlier it was demonstrated that the PRR is optimal for (a) the case in which all users desire only one relevant document and (b) the case where only two documents are retrieved. While experiencing such small information requirements or retrieved sets of documents is unusual, if such situations do occur the PRR would yield an optimal ranking strategy. (But, then the ORR would also yield the same result without any difficulty.)

Another alternative deals with the relationship between the numbers of documents and users. If the numbers of users are consistently larger than the numbers of documents, the difference in performance between the two ranking rules will be slight, as demonstrated in Figures 4-2 and 4-3. The PRR was found to have no more than a 10% loss as compared to the ORR when the numbers of users are at least three times the numbers of documents. (The data used to obtain this figure were simulated and, of course, would be more solid if the distribution of relevant documents among users were based upon observation.) Until a more efficient algorithm is developed to arrive at the optimal ranking procedure, the PRR would perform well if a three to one user-document ratio could be maintained among the retrieved documents.

Another alternative would be to improve the programmed version of the optimal algorithm. Several suggestions can be made in this respect:

1) The extensive use of disk input in the simulator program and disk input/output in the optimal ranking algorithm may be causing excessive CPU time. Replacing the file of random numbers by a random number generator would likely improve efficiency.

2) Using strings to store arrays, although maximizing space utilization, is probably causing excessive CPU time to pack and unpack the strings. At eleven documents only 10% of core capacity was being used, but 9 minutes of CPU time was needed for solving the problem.

3) The searching technique used to locate the permutations of all sequences at each stage of the optimal algorithm could no doubt be improved. According to Whitehead, this one factor can place extravagant demands upon CPU time. [47, p. 69].

4. The problem of getting data into and out of the computational algorithm at each stage also needs special attention. The author opted to use disk I/O for this requirement. One question that needs to be considered is whether disk input/output is the best method. In this study a routine (see Appendix C) was programmed to analyze the magnitude of storage requirements needed at both the beginning and end of each stage. In the light of the core requirements at each stage, one wonders whether a better procedure exists (than that used) for feeding data from one stage to the next.

If the programmed version of the optimal algorithm could be made more efficient, considerable benefit would be realized by the users as the number of retrieved documents approaches a simple value of twenty (see chapter four).

Another alternative by which the computational complexity of the ORR may be bypassed would be to remove inhibiting constraints. Two possibilities come to mind. (1) Deleting the requirement that a stop-search point be available for each user-type would avoid the permutation problem attached to sequencing. (2) Disregarding user differences and dealing with a summary value judgment representing all users would avoid considerable computational complexity. While the degradation in retrieval effectiveness may be justified by the increased efficiency of either of these modifications, their effect outside the behavior of the PRR has not been studied.

A final alternative to the optimal ranking rule would be to replace

the algorithm with an efficient heuristic. For example, starting with the ranking produced by the PRR one could interchange pairs of documents according to some orderly scheme until no more such switches produce any improvement. While the result may only be a "local" optimum, it might be substantially better than probability ranking and much more practical than the optimal algorithm. Lin and Kernigham espouse this approach. [37].

The success (or failure) of each of the foregoing alternatives could be determined by adopting either of the two methods used in this report for measuring performance. First, disregarding the cost of invoking the alternative ranking procedures, one could simply compute the average search lengths for each procedure and compare these values with the corresponding search lengths under the ORR. Since the ORR yields the best possible document ranking, an absolute referent is available for comparison--albeit difficult to produce for large numbers of documents.

Secondly, the approach used to compare the difference between the ORR and PRR, as developed in chapter six, could be applied. In contrast to those results, however, removing the ORR as a referent also removes an absolute standard against which the alternate procedures could be compared and thus implies that the procedure will be evaluated independently on its own merits. Using this procedure, the difference in search length improvement is computed between each alternate ranking procedure and the PRR. Also, the difference between the effort of applying each procedure and the effort required for the PRR is computed. A linear combination (See chapter six) of both difference functions provides a basis for evaluating the net worth of each ranking procedure.

In Retrospect

A philosophical observation seems appropriate in concluding this report. Review authors in the Annual Review of Information Science and Technology consistently maintain that current research fails to apply existing theory where it should be applied. In addition, they claim that lack of recognition of the similarities between current problems and previous solutions leads to needless ambiguities in the language of documentation. [54, 55]. They observe that the literature appears diffuse and they identify this condition as being due (in part) to the presence of diverse descriptions used to describe similar concepts. This results from a multiplicity of disciplines overlapping common subject areas.

Ironically, both disadvantages and benefits occur in a multidisciplinary field. One disadvantage is that related concepts may be described in such variant language from one discipline to another that cross communication between the disciplines fails. This condition was highlighted by a personal experience of the author when he, having previously invented the backtracking method used in the optimal algorithm, discovered several months later that it is a standard approach in integer programming. This problem is also experienced in many other areas of the knowledge industry.

On the other hand, there sometimes exists essential cross fertilization within a field when communication permits the exchange of solutions and procedures from one discipline for application to the problems of another. When interdisciplinary communication exists both disciplines benefit.

Not only in information science does the need for good communication exist, but in other fields as well. The essential point is that efficient

and effective communication of ideas exists to help the user identify information he truly needs and at the same time relieves him of information which is simply distracting. It is hoped that this study will contribute in some way to improving the quality of information providing systems.

APPENDIX A

ALGOL Program for Determining the ORR

The following procedure for determining the optimal ranking rule was run on a DEC KI/10 computer under ALGOL-60, version 4.

```
PROCEDURE OPTIMAL (N,M,ROW,OPTSEQ,COSTSEQ,NSORT,DOT,SEQ,FLAG,COST,CHECK);

! THE PROCEDURE 'OPTIMAL.ALG' USES A BACKTRACKING METHOD FOR DETER-;
! MINING THE MINIMUM COST PATH.  SPECIFIC TO THIS APPLICATION, IT WILL;
! DETERMINE THE OPTIMAL SEARCH SEQUENCE OF A SET OF RETRIEVED DOCUMENTS;
! ON THE BASIS OF PRIOR ESTIMATES OF THEIR WORTH TO THE USERS.;
! DURING THE STAGING OPERATION ESSENTIAL DATA IS STORED ON DISK.;

! 'N' IS THE NUMBER OF DOCUMENTS.;
! 'M' IS THE NUMBER OF USERS.;
! 'ROW' IS AN N BY M STRING ARRAY CONTAINING THE USER RELEVANCE;
! JUDGMENTS FOR THE N DOCUMENTS.  (STRING VARIABLES ARE USED TO ;
! PERMIT PACKING ON A BIT LEVEL.);
! THE OPTIMAL SEQUENCE IS RETURNED IN 'OPTSEQ'.;
! THE RETURNED VALUE IN 'COSTSEQ' REPRESENTS THE AMOUNT OF NON-;
! RELEVANT MATERIAL SCANNED BY THE USERS UNDER THE OPTIMAL SEQUENCE.;

   ! SINCE THIS VERSION OF ALGOL DOES NOT RELEASE FOR REUSE THE CORE;
   ! FROM ONE PROCEDURE CALL TO THE NEXT, THE REMAINING PARAMETERS IN;
   ! THE LIST ARE SIMPLY THERE TO ACHIEVE MORE EFFICIENT UTILIZATION;
   ! OF CORE IN BOTH THIS PROCEDURE AND THE SIMULATOR PROGRAM.;

VALUE N,M; INTEGER N,M; INTEGER ARRAY NSORT,DOT;
STRING OPTSEQ,COSTSEQ;  STRING ARRAY ROW,SEQ,FLAG,COST,CHECK;
BEGIN
STRING WORK2, WORK3, NEWFLAG, WORKSEQ, BUFF;
INTEGER GATE, STAGE,22,J,K,JFLAG,COUNT,SUM,L1,L2,L3,L4,I,COSTF,COSTR,
TEMP;
GATE ← 0; STAGE ← 2; JFLAG ← 0;
COSTSEQ ← NEWSTRING(1,9); OPTSEQ ← NEWSTRING(N,7);
WHILE STAGE < N DO
   BEGIN
   INTEGER SIZE, SIZE2,Z,D1,D2,D3;
   PRINT (STAGE,1); BREAKOUTPUT;

! THE BEGINNING AND FINAL NO. OF VALUES TO BE GENERATED AT THIS STAGE.;

   LABEL0:  D3 ← 1; D2 ← 1;
   FOR D1 ← 1 UNTIL STAGE DO
```

```
      BEGIN
      D2 ← (N - D1 + 1) *D2;
      D3 ← (D1 * D3);
      END;
   IF JFLAG = 1 THEN GO TO LABELOO;
   SIZE ← D2/D3;
   JFLAG ← 1; STAGE ← STAGE + 1; GO TO LABELO;
   LABELOO:  SIZE2 ← D2/D3; JFLAG ← 0; STAGE ← STAGE -1;
      BEGIN
      COUNT ← SUM ← L1 ← L2 ← COSTF ← COSTR ← Z ← L ← I ← L3 ← L4 ← 0;
      TEMP ← 0;
      COSTF ← COSTR ← 0; Z ← 1;
      DELETE (COSTSEQ); COSTSEQ ← NEWSTRING (1,9);
      NEWFLAG ← NEWSTRING (M,1);
      IF GATE = 1 THEN GO TO LABEL1;

! GENERATE PAIRS;

      FOR I←1 UNTIL N-1 DO
         BEGIN
         FOR J ← I+1 UNTIL N DO
            BEGIN
            SEQ[Z] ← NEWSTRING (N,4); FLAG[Z] ← NEWSTRING (M,1);
            COST[Z] ← NEWSTRING (1,9); CHECK[Z] ← NEWSTRING (N,1);
            FOR K ← 1 UNTIL M DO
              BEGIN
! SET USER FLAG TO LATER DETERMINE COST FUNCTION;

              TEMP ← ROW[I].[K] + ROW[J].[K];
              IF TEMP > 0 THEN FLAG [Z].[K] ← 1;
              TEMP ← ROW[I].[K] - ROW [J].[K];
              IF TEMP < 0 THEN COSTF ← COSTF - TEMP
              ELSE IF TEMP > 0 THEN COSTR ← COSTR + TEMP;
              END K LOOP;

! COMPARE COST OF PAIR WITH ITS INVERSE AND PRESERVE CHEAPER ROUTE;

            IF COSTF < = COSTR THEN
              BEGIN
              SEQ[Z].[N] ← J;  SEQ[Z].[N-1] ← I;
              COST[Z].[1] ← COSTF;
              END
            ELSE
              BEGIN
              SEQ[Z].[N] ← I;  SEQ[Z].[N-1] ← J;
              COST[Z].[1] ← COSTR;
              END;
            COSTR ← COSTF ← 0; Z ← Z + 1;
            END J LOOP;
      END I LOOP;
      LABEL1:  IF GATE = 0 THEN GO TO LABEL2;
```

```
! LOAD PREVIOUS STAGE FROM DISK INTO ARRAYS;

      INPUT (6, "DSK",12,5); SELECTINPUT (6); OPENFILE (6,"STAGE.DAT");
      FOR I ← 1 UNTIL SIZE DO
         BEGIN
         SEQ[I] ← NEWSTRING (N,4); FLAG[I] ← NEWSTRING (M,1);
         COST[I] ← NEWSTRING (1,9); CHECK[I] ← NEWSTRING (N,1);
         FOR J ← 1 UNTIL N DO
            BEGIN
            INSYMBOL(K); SEQ[I].[J] ← K;
            END J LOOP;
         FOR J ← 1 UNTIL M DO
            BEGIN
            INSYMBOL(K); FLAG[I].[J] ← K;
            END I LOOP;

! DELETE OLD FILE AND OPEN NEW FILE;

      OPENFILE (6,""); RELEASE (6);
      LABEL2:  GATE ← 1; STAGE ← STAGE + 1; Z2 ← 1;
      OUTPUT (6,"DSK",12,5); SELECTOUTPUT (6); OPENFILE (6,"STAGE.DAT");

! IDENTIFY FEASIBLE NUMBERS NOT YET IN MAIN SEQUENCE;

      FOR I ← 1 UNTIL SIZE - 1 DO
         BEGIN
         Z ← 1;
         FOR J ← 1 UNTIL N DO DOT[J,2] ← 0;
         FOR K ← 1 UNTIL N DO
            BEGIN
            FOR J ← 1 UNTIL STAGE - 1 DO
              BEGIN
              IF SEQ[I].[N-J+1] = K THEN GO TO LABEL3;
              END J LOOP:
            NSORT[Z] ← K; Z ← Z +1; GO TO LABEL4;
            LABEL3:  DOT[K,2] ← 1;
            LABEL4:  END K LOOP;

! EXPLORE THE NEXT STAGE OF MAIN SEQUENCE;

         FOR K ← 1 UNTIL Z-1 DO
            BEGIN
            L1 ← NSORT[K];
            IF CHECK[I].[L1]  = 1 THEN GO TO LABEL8;
            COUNT ← 0; COSTF ← 0;
            WORKSEQ ← COPY(SEQ[I]);
            WORKSEQ.[N-STAGE+1] ←  NSORT [K]; DOT[NSORT[K],2] ← 1;

! COMPUTE COST OF NEW MAIN SEQUENCE;

            FOR 1 ← 1 UNTIL M DO
              BEGIN
              NEWFLAG.[L] ← 0;
```

```
            IF FLAG[I].[L] = 1 AND ROW [L1].[L] = 0 THEN
               COSTF ← COSTF + 1;
            IF FLAG[I].[L] = 1 OR ROW[L1].[L] = 1 THEN
               NEWFLAG.[L] ← 1;
            END;
         COSTF ← COSTF + COST[I].[1];
         FOR L ← I+1 UNTIL SIZE DO
           BEGIN

! FORM MASK FOR NEXT SEQUENCE TO BE COMPARED WITH MAIN SEQUENCE:

            FOR L3 ← 1 UNTIL N DO DOT[L3,1] ← 0;
            FOR L4 ← N STEP -1 UNTIL N - STAGE +2 DO
               BEGIN
               L2 ← SEQ[L].[L4];
               DOT[L2,1] ← 1;
               END;
            SUM ← 0;

! COMPUTE OVERLAP BETWEEN TWO SEQUENCES;

            FOR L4 ← 1 UNTIL N DO
               SUM ← SUM + DOT[L4,1] * DOT[L4,2];
            IF SUM = STAGE - 1 THEN
               BEGIN

! DETERMINE MISSING NUMBER FROM SEQUENCE IN QUESTION;

               WORK2 ← COPY (SEQ[L]);
               COUNT ← COUNT + 1;


                  FOR L4 ← 1 UNTIL STAGE DO
                     BEGIN
                     FOR L3 ← 1 UNTIL STAGE - 1 DO
                        BEGIN
                        IF WORKSEQ.[N-L4+1]=WORK2.[N-L3+1] THEN
                           GO TO LABEL5;
                        END L3 LOOP;
                     L2 ← WORKSEQ.[N-L4+1]
                     WORK2.[N-STAGE+1] ← L2;
                     CHECK[L].[L2] ← 1;
                     GO TO LABEL6;
                     LABEL5:  END L4 LOOP;

! COMPUTE COST OF SEQUENCE IN QUESTION;

               LABEL6:  WORK3 ← NEWSTRING (M,1); COSTR ← 0;
               FOR L3 ← 1 UNTIL M DO
                  BEGIN
                  L1 ← WORK2.[N-STAGE + 1];
                  IF FLAG[L].[L3] = 1 AND ROW[L1].[L3] = 0
                     THEN COSTR ← COSTR + 1;
```

```
                    IF FLAG[L].[L3] = 1 OR ROW[L1].[L3] = 1
                       THEN WORK3.[L3] ← 1;
                    END L3 LOOP;
                 COSTR ← COSTR + COST[L].[1];

! DETERMINE LEAST COST SEQUENCE AND CALL THIS THE MAIN SEQUENCE;

                 IF COSTF > COSTR THEN
                    BEGIN
                    COSTF ← COSTR;
                    DELETE (WORKSEQ); DELETE (NEWFLAG);
                    WORKSEQ ← COPY (WORK2); NEWFLAG ← COPY (WORK3);
                    END;
                 DELETE (WORK2); DELETE (WORK3);
                 END STAGE BLOCK;
              IF COUNT = STAGE - 1 THEN GO TO LABEL7;
              END L LOOP;
           LABEL7:  COSTSEQ.[1] ← COSTF;

! CONVERT DATA TO ASCII CODE FOR DISPLAY.;

           FOR L3 ← 1 UNTIL N DO OPTSEQ.[L3] ← WORKSEQ.[L3] + 48;
           IF STAGE = N THEN
              BEGIN
              TEMP ← 99999;
              GO TO LABEL9;
              END;

! STORE RESULTS FROM THIS STAGE ONTO DISK.;

           FOR L3 ← 1 UNTIL N DO OUTSYMBOL (WORKSEQ.[L3]);
           FOR L3 ← 1 UNTIL M DO OUTSYMBOL (NEWFLAG.[L3]);
           OUTSYMBOL (COSTSEQ.[1]); Z2 ← Z2 + 1;
           DOT[NSORT[K],2] ← 0;
           LABEL8:  IF Z2 - 1 = SIZE2 THEN GO TO LABEL9;
           END K LOOP;
        END I LOOP;
        LABEL9:  DELETE (WORKSEQ); DELETE (NEWFLAG);
     END DYNAMIC DIMENSION BLOCK;
  OUTSYMBOL (COSTSEQ.[1]);
  RELEASE (6);
  FOR I ← 1 UNTIL SIZE DO
     BEGIN
     DELETE (SEQ[I]); DELETE (FLAG[I]);
     DELETE (COST[I]); DELETE (CHECK[I]);
     END;
  IF TEMP = 99999 THEN GO TO LABEL10;
  END STAGE BLOCK;
LABEL10:  END
```

APPENDIX B

ALGOL Program for Simulating Document Retrieval

The following simulation program was run on a DEC K1/10 computer under ALGOL-60, version 4.

```
BEGIN
! THE FOLLOWING PROGRAM SIMULATES A DOCUMENT RETRIEVAL SYSTEM BY;
! GENERATING VARIABLY DIMENSIONED USER-DOCUMENT MATRICES.  FOR EACH;
! USER BOTH THE NUMBER AND THE CHOICE OF RELEVANT DOCUMENTS ARE;
! RANDOMLY DETERMINED.  THIS INFORMATION IS STORED IN THE ARRAY 'ROW'.;
! GIVEN EACH MATRIX, PERFORMANCE MEASURES FOR BOTH PROBABILITY AND;
! OPTIMAL RANKING ARE COMPARED.  THE EXTERNAL PROCEDURE FOR DETERMINING;
! THE OPTIMAL RANKING IS CALLED 'OPTIMAL'.  RANDOM NUMBERS ARE OBTAINED;
! FROM A FILE NAMED 'RANDOM.DAT'.  STRING ARRAYS ARE USED SO THAT;
! PACKING MAY BE ACHIEVED AT THE BIT LEVEL.;

! SINCE THIS VERSION OF ALGOL DOES NOT AUTOMATICALLY RELEASE STORAGE,;
! NUMEROUS 'DELETE' COMMANDS ARE USED.  IN ADDITION ALL OF THE MEMORY;
! SPACE REQUIRED IN THE EXTERNAL PROCEDURE 'OPTIMAL' IS ALLOCATED IN;
! THIS PROGRAM AND PASSED AS PARAMETERS IN THE CALLING PROCEDURE.;

INTEGER MDOC,MUSER,AMAX,NULL;
WRITE ("INPUT THE NUMBER OF DOCUMENTS"); NEWLINE; READ (MDOC);
WRITE ("INPUT THE NUMBER OF USERS"); NEWLINE; READ (MUSER);
WRITE ("INPUT THE MAX ARRAY SIZE"); NEWLINE; READ (AMAX);
WRITE ("INPUT THE START POINT IN RANDOM TABLE"); NEWLINE; READ (NULL);
   BEGIN
   EXTERNAL PROCEDURE OPTIMAL;
   STRING STRTEM,OPTSEQ,COSTSEQ;
   STRING ARRAY ROW[1:MDOC];
   STRING ARRAY SEQ,FLAG,COST,CHECK[1:AMAX];
   INTEGER I, I1, I2, I3, I4, I5, I6, J, J1, ICOUNT,NM,M2,N5,USER,LENGTH;
   INTEGER NRAN,NRAN2,NSUM,NTEMP,NRSUM,NOTSUM,JFLAG,JDOC,L1,L2,L3,L7,MAX;
   INTEGER S, K2,K,L4;
   INTEGER ARRAY NDOC[1:MUSER], NSORT[1:MDOC], DOT[1:MDOC,1:2];
   INPUT (5,"DSK"); SELECTINPUT (5); OPENFILE (5,"RANDOM.DAT");

   ! SET STARTING LOCATION IN RANDOM NUMBER TABLE;

   FOR I ← 1 UNTIL NULL DO INSYMBOL (I1);
   NM ← 100; M2 ← 1; I ← 1;
   FOR N5 ← 1 UNTIL 4 DO
      BEGIN
      ICOUNT ← 0;
      FOR I3 ← 1 UNTIL MDOC DO ROW[I3] ← NEWSTRING(MUSER,1);
      FOR USER ← 1 UNTIL MUSER DO
```

```
! DETERMINE NUMBER OF RANDOM DIGITS NEEDED;

   BEGIN
   LENGTH ← LN (MDOC)/LN (10) + 1&-05 + 1;
   LENGTH ← LENGTH DIV 1;

! RANDOMLY SELECT NUMBER OF REL DOCS FOR EACH USER, STORE IN 'NDOC';
   LOOP7: NRAN ← 0;
   FOR I3 ← 1 UNTIL LENGTH DO
      BEGIN
      INSYMBOL (S);
      IF S = 69 THEN
         BEGIN
         WRITE ("RANDOM FILE EXHAUSTED");
         NEWLINE; GO TO STOP;
         END;
! CHARACTER TO INTEGER CONVERSION;

   NRAN ← NRAN + (S-48) * 10 ↑ (LENGTH-I3);
   ICOUNT ← ICOUNT + 1;
   END;

! TEST RANGE OF RANDOM NUMBER;

   IF NRAN > MDOC-1 OR NRAN <  0 THEN GO TO LOOP7;
   NDOC[USER] ← (NRAN + 1);

! NDOC[USER] CONTAINS THE NO. RELEVANT DOCS FOR EACH USER;
! RANDOMLY SELECT (WITHOUT REPLACEMENT) WHICH OF THE DOCS IS  RELEVANT;

   FOR K ← 1 UNTIL NDOC[USER] DO
      BEGIN
      LENGTH ← LN (NDOC[USER])/LN (10) +1.00001;
      LENGTH ← LENGTH DIV 1;
      LOOP8:  NRAN ← 0;
      FOR I3 ← 1 UNTIL LENGTH DO
         BEGIN
         INSYMBOL(S);
         IF S = 69 THEN
           BEGIN
           WRITE ("RAND FILE EXHAUSTED");
           NEWLINE; GO TO STOP;
           END;
         NRAN ← NRAN + (S-48) * 10 ↑ (LENGTH-I3);
         ICOUNT ← ICOUNT + 1;
         END;
      IF NRAN > (MDOC-K) OR NRAN <  0 THEN GO TO LOOP8;
      NRAN2 ← NRAN + 1;
      I5 ← 1; I6 ← 0;
      LOOP1:  IF ROW[I5].[USER] = 0 THEN
      I6 ← I6 + 1;
      IF NRAN2 = I6 THEN
         BEGIN
```

```
! 'ROW' ARRAY IS THE USER-DOCUMENT MATRIX;
! IF DOC IS RELEVANT THEN ROW ELEMENT IS EQUAL TO "1";

            ROW[I5].[USER] ← 1;
            GO TO ENLOOP;
            END
         ELSE
            BEGIN
            I5 ← I5 +1; GO TO LOOP1;
            END;
         ENLOOP:  END;
      END USER LOOP;
   NEWLINE(2);

! SORT DOCS BY DECREASING FREQUENCY

   FOR L1 ← 1 UNTIL MDOC DO
      BEGIN
      NSUM ← 0;
      FOR L2 ← 1 UNTIL MUSER DO
      NSUM ← NSUM + ROW[L1].[L2];
       NSORT[L1] ← NSUM * 10 + 3 +L1;
      END;
   MAX ← MDOC;
   LOOP2:  JFLAG ← 0;
   FOR L3 ← 2 UNTIL MAX DO
      BEGIN
      IF NSORT[L3-1] < NSORT[L3] THEN
         BEGIN
         NTEMP ← NSORT[L3-1];
         NSORT[L3-1] ← NSORT[L3];
         NSORT[L3] ← NTEMP;
         STRTEM ← COPY (ROW[L3-1]);
         DELETE (ROW[L3-1]);
         ROW[L3-1] ← COPY (ROW[L3]);
         DELETE (ROW[L3]);
         ROW[L3] ← COPY (STRTEM);
         DELETE (STRTEM);
         JFLAG ← 1;
         END;
      END L3 LOOP;
   IF JFLAG = 1 THEN
      BEGIN
      MAX ← MAX - 1; GO TO LOOP2;
      END;

   ! COMPUTE  PERFORMANCE MEASURES FOR FREQUENCY RANKED DOCS;

   NRSUM ← 0; NOTSUM ← 0;
   FOR L3 ← 1 UNTIL MUSER DO
      BEGIN
      NRSUM ← NRSUM + NDOC[L3];
      FOR L4 ← MDOC STEP -1 UNTIL 1 DO
```

```
        BEGIN
        IF ROW[L4].[L3] = 1 THEN
           BEGIN
           NOTSUM ← NOTSUM + L4 - NDOC[L3];
           GO TO LOOP5; END;
        ELSE GO TO LOOP4;
        LOOP4:  END OF L4 LOOP;
     LOOP5:  END;
   WRITE ("COST OF FREQUENCY RANKING"); PRINT (NOTSUM);
NEWLINE:

! OPTIMALLY RANK DOCUMENTS
   OPTIMAL (MDOC, MUSER,ROW,OPTSEQ,COSTSEQ,NSORT,DOT,SEQ,FLAG,COST,CHECK);
   SELECTINPUT (5);

! COMPUTE PERFORMANCE MEASURE FOR THE OPTIMAL DOC RANKING;

   L3 ← COSTSEQ.[1]; NEWLINE; WRITE ("OPTSEQ ="); WRITE (OPTSEQ);
   WRITE (" COST = "); PRINT (L3); NEWLINE;
   DELETE (OPTSEQ); DELETE (COSTSEQ);
   FOR L3 ← 1 UNTIL MDOC DO DELETE (ROW[L3]);
   WRITE ("NO. RANDOM #'S USED ="); PRINT (ICOUNT); NEWLINE;
   END N5 LOOP;
RELEASE (5);
STOP:  END DYNAMIC DIMENSION BLOCK;
END
```

APPENDIX C

Program for Determining Magnitude of Storage Requirements During and at End of Each Stage of the ORR

The following program for determining the magnitude of storage requirements both during and at the end of each stage under the backtracking procedure used by the optimal ranking algorithm was run on a DEC K1/10 computer under ALGOL-60, version 4.

```
BEGIN
INTEGER I, J, K, K1, Z;
REAL PRODUCT1, PRODUCT2, SUM;
FOR I ← 3 UNTIL 25 DO
   BEGIN
   WRITE (" NUMBER OF DOCUMENTS ="); PRINT (I); NEWLINE (2);
   SUM ← 0;
   FOR J ← I-1 STEP-1 UNTIL 1 DO
      BEGIN
      PRODUCT1 ← 1; PRODUCT2 ← 1; Z ← 1;
      FOR K ← I STEP-1 UNTIL J DO
         BEGIN
         PRODUCT1 ← PRODUCT1 * K/Z;
         IF K<I THEN Z ← Z+1;
         PRODUCT2 ← PRODUCT2 * K/Z;
         END;
      K ← I-J; K1 ← K+1;
      WRITE ("stage:"); PRINT (K1); NEWLINE;
      WRITE ("NO. OF PATHS IN THIS STAGE..."); PRINT (PRODUCT1);
      NEWLINE;
      WRITE ("NO. OF STORAGE LOCATIONS AT END OF STAGE...");
      PRINT (PRODUCT2); NEWLINE;
      SUM ← SUM + PRODUCT1;
      END;
   WRITE ("TOTAL NO. OF PATHS..."); PRINT (SUM);
   NEWLINE (3);
   END;
END
```

APPENDIX D

Application of Equation (6), Chapter Three

To illustrate the application of equation (6), chapter three, the example portrayed in Figure 2-4 of chapter two will be used.

		User									
		1	2	3	4	5	6	7	8	9	
	6		x			x			x		1
	2	x	x	x	x	x					2
	1	x	x	x	x	x					3
Document	5	x			x			x			4 Row
	7			x			x			x	5
							x	x	x	x	6
	4						x	x	x	x	7

Figure 2-4, Chapter 2

$$SL = \sum_{i=2}^{7} \left[\sum_{j=1}^{9} |Q_{i,j}| \ (i-1) - \sum_{j=1}^{9} |Q_{k_j,j} \cap Q_{i,j}| \ (k_j) \right] \qquad (6)$$

$$=(5)(1)-(0)(0)-(1)(1)-(0)(0)-(0)(0)-(1)(1)-(0)(0)-(0)(0)-(0)(1)-(0)(0)$$

$$+(5)(2)-(1)(2)-(1)(2)-(1)(2)-(1)(2)-(1)(2)-(0)(0)-(0)(0)-(0)(1)-(0)(0)$$

$$+(3)(3)-(1)(3)-(0)(3)-(0)(3)-(1)(3)-(0)(3)-(0)(0)-(0)(0)-(0)(1)-(0)(0)$$

$$+(3)(4)-(0)(4)-(0)(3)-(1)(3)-(0)(4)-(0)(3)-(0)(0)-(0)(4)-(0)(1)-(0)(0)$$

$$+(4)(5)-(0)(4)-(0)(3)-(0)(5)-(0)(4)-(0)(3)-(1)(5)-(1)(4)-(1)(1)-(1)(5)$$

$$+(4)(6)-(0)(4)-(0)(3)-(0)(5)-(0)(4)-(0)(3)-(1)(6)-(1)(6)-(1)(6)-(1)(6)$$

$$= \begin{array}{l} 5-2 \\ +10-10 \\ +\ 9-6 \\ +12-3 \\ +20-15 \\ +24-24 \end{array} = 3 + 0 + 3 + 9 + 5 + 0 = 20$$

For each row the first expression in equation (6) acts as if the relevant documents on the i^{th} row are the first encountered by the respective users and thus implies that each user has just encountered (i-1) nonrelevant documents. However, the second expression - and in particular $|Q_{k_j,j} \cap Q_{i,j}|$, acts like a binary switch in determining the position of the previous relevant document for each user. If located, the row value (i.e., k_j) of that document is subtracted from (i-1).

Note that there exists a row of computations for the second through the seventh documents. Equation (6) has no penalty for reading the first document since if the document is relevant there obviously would be none and if the document is nonrelevant to the user its penalty will not be assigned to him until he has reached his first relevant document. If the user has no relevant documents in the retrieved set then the discussion on page 40, chapter three should be considered.

APPENDIX E

Developing Equation (7) of Chapter Two Under the Case II Assumptions

Analogous to the procedure used in the Case I proof of chapter two we construct a document ranking based upon a priori evaluation of N documents by M users. We assume that the optimal sequence will follow that of the PRR. That is, if $\sum_{j=1}^{M} |Q_{s,j}|$ represents the number of patrons who find the s^{th} document relevant to their common request, then it is proposed that the optimal sequence is that ranking for which the cardinality of relevance assessments is strictly decreasing. Where the Case I proof dealt exclusively with all users achieving their stop-search points after obtaining one and only one relevant document, the Case II proof permits each user to have any number of relevant documents associated with his stop-search point. This departure requires that the expression for search length be modified from that of equation (1), chapter two, to equation (6) - which is reproduced here:

$$SL = \sum_{i=2}^{N} \left[\sum_{j=1}^{M} | Q_{i,j} | (i-1) - \sum_{j=1}^{M} | Q_{k_j,j} \cap Q_{i,j} | (kj) \right] \qquad (E\text{-}1)$$

where $k_j = \left\{ \begin{matrix} \max(k): & Q_{k,j} \cap Q_{i,j} \neq \emptyset \ \forall \ k < i \\ 0: & Q_{k,j} \cap Q_{i,j} = \emptyset \ \forall \ k < i \end{matrix} \right\}$

Proceeding on the basis that $\sum_{j=1}^{M} | Q_{s,j} | > \sum_{j=1}^{M} | Q_{s+1,j} | \quad \forall \ s$, except

in the one instance where $\sum_{j=1}^{M} |Q_{i,j}| < \sum_{j=1}^{M} |Q_{i+1,j}|$, we determine the search length for two almost identical document sequences. One sequence conforms to the condition of strict decreasing cardinality of the PRR, and the other breaks from decreasing cardinality by interchanging the i^{th} and i+1 documents. Several considerations will simplify and hopefully clarify the following development. Since our aim is to obtain the difference between the two search lengths, the series of terms in the search length expressions preceding the i^{th} and $i+1^{th}$ documents will be identical for both sequences and, consequently, upon subtracting one sequence from the other, will cancel each other. For both sequences these preceding terms will simply be identified by "A". In addition, superscripts representing their actual positions in the sequence, will be assigned to $Q_{i,j}$ and $Q_{i+1,j}$. (Even when documents are interchanged in a sequence, the relative position number (i.e., row values) will remain invariant - i.e., 1, 2, 3,..., i-1, i, i+1,..., N-1, N). The notation, then, $Q^{i}_{i+1,j}$ reflects to the j^{th} user who judged a document relevant which, originally, was in the i+1 position, but has now been moved to the i^{th} position in the sequence.

Observing the condition placed upon k_j in equation (E-1), the superscripts assigned to $Q_{k_j,j}$ will not have exact values as do the preceding ones. In english, the condition on k_j simply means - assign the sequence position to k_j at which the j^{th} user encounters a relevant document that is preceding but nearest to the i^{th} position. Since we will not know this value without observing the user-document matrix in question, we will adopt the convention that the superscript represents the largest row value that the term may be assigned. Thus,

the notation $Q_{k_j,j}^{\leq i-1}$ refers to the j^{th} user who has determined a document in position k_j to be relevant where k_j is the largest value consistent with $k_j \leq (i-1)$. In what follows k_j will sometimes appear as a separate term, itself. To record the knowledge we may have that, for example, k_j cannot assume a position value greater than i-1, its superscript notation will be specified as $k_j^{\leq i-1}$.

As a final housekeeping note, if a prime (apostrophe) is attached to a notation, this signifies that the expression is associated with a search length whose documents are ranked according to the PRR.

The search length for decreasing cardinality becomes

$$SL' = A + \sum_{j=1}^{M} |Q_{i+1,j}|(i-1) - \sum_{j=1}^{M} |Q_{k_j,j}^{\leq i-1} \cap Q_{i+1,j}^{i}|(k_j^{\leq i-1})'$$

$$+ \sum_{j=1}^{M} |Q_{i,j}|\,(i) - \sum_{j=1}^{M} |Q_{k_j,j}^{\leq i} \cap Q_{i,j}^{i+1}|\,(k_j^{\leq i})'$$

$$+ \ldots$$

$$+ \sum_{j=1}^{M} |Q_{N,j}|\,(N-1) - \sum_{j=1}^{M} |Q_{k_j,j}^{\leq N-1} \cap Q_{N,j}^{N}|\,(k_j^{\leq N-1})' \qquad \text{(E-2)}$$

where $(k_j^{\leq i-1})' \leq i-1$, $(k_j^{\leq i})' \leq i$, ... , $(k_j^{\leq N-1})' \leq N-1 \quad \forall k_j$

and are subject to the maximization condition, in general,

$$(k_j^{\leq p-1})' = \begin{bmatrix} \max k_j' : & Q_{k_j',j} \cap Q_{p,j} \neq \emptyset \\ 0, \text{ if } & Q_{k_j',j} \cap Q_{p,j} = \emptyset \end{bmatrix} \qquad \forall j: \; k_j' \leq p-1$$

Inverting $\sum_{j=1}^{M} |Q_{i,j}|$ and $\sum_{j=1}^{M} |Q_{i+1,j}|$ under the proposal that a shorter search length will be achieved we have

$$SL = A + \sum_{j=1}^{M} |Q_{i,j}| \, (i-1) - \sum_{j=1}^{M} |Q_{k_j,j}^{\leq i-1} \cap Q_{i,j}| \, (k_j^{\leq i-1})$$

$$+ \sum_{j=1}^{M} |Q_{i+1,j}| \, (i) - \sum_{j=1}^{M} |Q_{k_j,j}^{\leq i} \cap Q_{i+1,j}^{i+1}| \, (k_j^{\leq i})$$

$$+ \ldots$$

$$+ \sum_{j=1}^{M} |Q_{N,j}| \, (N-1) - \sum_{j=1}^{M} |Q_{k_j,j}^{\leq N-1} \cap Q_{N,j}^{N}| \, (k_j^{\leq N-1}) \qquad \text{(E-3)}$$

subject to the constraints on $k_j^{\leq i-1}$, $k_j^{\leq i}$, ... , $k_j^{\leq N-1}$.

Because of the inversion of documents i and i+1, the value of $k_j^{\leq N-1}$ at the N^{th} document in the sequence is not necessarily equal for the two search lengths SL′ and SL. Hence the need for the prime is restated - to differentiate between $(k_j^{\leq N-1})'$ and $(k_j^{\leq N-1})$. Subtracting the two search expressions (equations (E-2) and (E-3)) we have

$$SL' - SL = \sum_{j=1}^{M} |Q_{i,j}| - \sum_{j=1}^{M} |Q_{i+1,j}|$$

$$+ \sum_{j=1}^{M} |Q_{k_j,j}^{\leq i-1} \cap Q_{i,j}^{i}| \, (k_j^{\leq i-1}) - \sum_{j=1}^{M} |Q_{k_j',j}^{\leq i} \cap Q_{i,j}^{i+1}| \, (k_j^{\leq i})'$$

$$+ \sum_{j=1}^{M} |Q_{k_j,j}^{\leq i} \cap Q_{i+1,j}^{i+1}| \, (k_j^{\leq i}) - \sum_{j=1}^{M} |Q_{k'_j,j}^{\leq i-1} \cap Q_{i+1,j}^{i}| \, (k_j^{\leq i-1})'$$

$$+ \ldots$$

$$+ \sum_{j=1}^{M} |Q_{k_j,j}^{\leq N-1} \cap Q_{N,j}^{N}| \, (k_j^{\leq N-1}) - \sum_{j=1}^{M} |Q_{k'_j,j}^{\leq N-1} \cap Q_{N,j}^{N}| \, (k_j^{\leq N-1})' \tag{E-4}$$

Several considerations will simplify the interpretation of this expression. First, all terms from $i+2$ to N can be represented by the expression

$$\sum_{\ell=i+2}^{N} \sum_{j=1}^{M} \left[|Q_{k_j,j}^{\leq \ell-1} \cap Q_{\ell,j}^{\ell}| \, (k_j^{\leq \ell-1}) - |Q_{k'_j,j}^{\leq \ell-1} \cap Q_{\ell,j}^{\ell}| \, (k_j^{\leq \ell-1})' \right] \tag{E-5}$$

This is an important result, and more will be said about it later.

Next, looking at the third and fourth expressions on the right hand side of the equation (E-4), we note that because $k_j^{\leq i-1}$ may assume values not greater than $i-1$ while $k_j^{\leq i}$ may assume values as great as i, in general, $(k_j^{\leq i-1}) \leq (k_j^{\leq i})'$, so the combined effect of these two terms will be ≤ 0. Similarly, for the fifth and sixth terms, in general, $(k_j^{\leq i}) \geq (k_j^{\leq i-1})'$, so that the combined effect of the latter two terms will be ≥ 0. In fact, the net effect of all four terms will be 0.

Proof

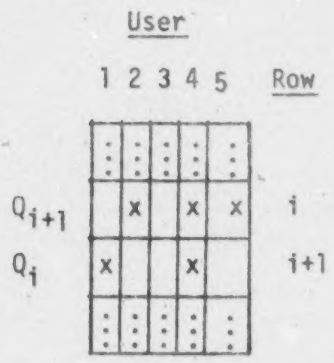
Assume the cardinally decreasing configuration as in Figure (E-1) which represents all possible configurations of document combinations in a two document sequence. Invert the sequence as in Figure (E-2) and

with both configurations compute L, the net search length associated with the third, fourth, fifth, and sixth terms of equation (E-4).

$$L = \sum_{j=1}^{M} |Q^{\leq i-1}_{k_j,j} \cap Q^{i}_{i,j}| \, (k_j^{\leq i-1}) - \sum_{j=1}^{M} |Q^{\leq i}_{k_j,j} \cap Q^{i+1}_{i,j}| \, (k_j^{\leq i}) +$$

$$+ \sum_{j=1}^{M} |Q^{\leq i}_{k_j,j} \cap Q^{i+1}_{i+1,j}| \, (k_j^{\leq i}) - \sum_{j=1}^{M} |Q^{\leq i-1}_{k_j,j} \cap Q^{i}_{i+1,j}| \, (k_j^{\leq i-1}) \quad \text{(E-6)}$$

Observing the superscripts on the Q's we note that equation (E-6) pertains only to documents either preceeding or at the $i+1$ row in the sequence. Although we do not know which documents prior to the i^{th} row are relevant and the $Q^{\leq i}_{k_j,j}$ requires this information, we will simply proceed with the notation of equation (E-6) unless Figures (E-1 and E-2) give us specific row values. Follow the convention that the normal search sequence is downward, and, again, an "x" means that the document is relevant.



(Decreasing Cardinality)

Figure E-1

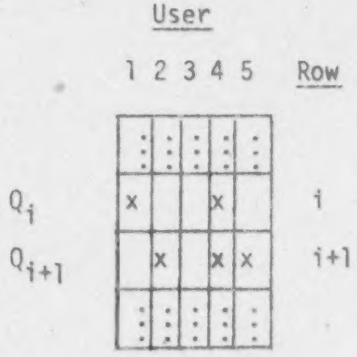


Figure E-2

All possible configurations of relevant documents in a two-document sequence.

$$L = (1)\ (k_1^{\leq i-1}) + (0)\ (k_2^{\leq i-1}) + (0)\ (k_3^{\leq i-1}) + (1)\ (k_4^{\leq i-1}) + (0)\ (k_5^{\leq i-1})$$

$$- [(1)(k_1^{\leq i}) + (0)(i) + (0)(k_3^{\leq i}) + (1)(i) + (0)(i)]'$$

$$+ (0)(i) + (1)(k_2^{\leq i}) + (0)(k_3^{\leq i}) + (1)(i) + (1)(k_5^{\leq i})$$

$$- [(0)(k_1^{\leq i-1}) + (1)(k_2^{\leq i-1}) + (0)(k_3^{\leq i-1}) + (1)(k_4^{\leq i-1}) + (1)(k_5^{\leq i-1})]'$$

$$= (k_1^{\leq i-1}) + (k_4^{\leq i-1}) - (k_1^{\leq i})' - (i)' + (k_2^{\leq i}) + (i) + (k_5^{\leq i}) - (k_2^{\leq i-1})'$$

$$- (k_4^{\leq i-1})' - (k_5^{\leq -1})' \qquad \text{(E-7)}$$

Referring to Figure (E-1), we note that for user #1 there is no relevant document in the i^{th} row. Hence, the superscript inequality of the third expression in equation (E-7) can be changed to $\leq (i-1)$. Similarly, for user #2 and user #5 in Figure (E-2), there are no relevant documents in the i^{th} row, so both superscript inequalities in the fifth and seventh expressions of equation (E-7) can be changed to $\leq (i-1)$. The distinction between $(k_j^{\leq i-1})'$ and $(k_j^{\leq i-1})$ may now be dropped since they both refer to the same row value. Thus, $L = 0$.

Since the above configuration satisfies $\sum^{M} |Q_{i+1,j}| > \sum^{M} |Q_{i,j}|$ and the example used contains all possible combinations of user judgments for two consecutive documents, any increase in the number of users will simply result in a duplication of one of the user patterns but will still obtain a net result of zero.

That L = 0 is not strange since the four expressions are simply a measure of the differences in search length for each user with respect to any relevant document at the i and i+1 position in the sequence and a previous relevant document. Inverting documents i and i+1 does not change the position [i.e., the value of $(k_j^{\leq i-1})$ or $(k_j^{\leq i-1})'$] of the previous relevant document for any user whether or not either, both, or none of the i and i+1 documents are relevant. This ends the proof.

Omitting equation (E-6) from equation (E-4) and using the simplification of expression (E-5), equation (E-4) becomes

$$SL' - SL = \sum_{j=1}^{M} |Q_{i,j}| - \sum_{j=1}^{M} |Q_{i+1,j}|$$

$$+ \sum_{\ell=i+2}^{N} \sum_{j=1}^{M} \left| |Q_{k_j,j}^{\leq \ell-1} \cap Q_{\ell,j}^{\ell}|(k_j^{\leq \ell-1}) - |Q_{k_j'}^{\leq \ell-1} \cap Q_{\ell,j}^{\ell}| (k_j^{\leq \ell-1})' \right| \qquad \text{(E-8)}$$

where the conditions on k_j are the same as those following equation (E-2). Equation (E-8) represents the expression sought in equation (7) of Chapter Three.

APPENDIX F

Developing Equation (6) of Chapter Five

The staging algorithm defined in chapter five will at any stage k ($1 \leq k \leq N-1$) for N document numbers begin with $\binom{N}{k} = \frac{N(N-1)\ldots(N-k+1)}{k!}$ different combinations of partial sequences--each of which has a cost function. From these will be generated $\binom{N}{k}(N-k)$ partial sequences having one additional document number. After eliminating the permuted partial sequences which violate the minimum cost function, the final number of partial sequences at each stage will be $\binom{N}{k+1}$ --the latter amount representing the initial number of sequences for the next stage. The total number of cost functions which are computed by the staging algorithm for a given number of documents, N, and M users is:

$$\text{total cost comparisons} = M \sum_{k=1}^{N-1} \binom{N}{k}(N-k) = M \sum_{k=1}^{N-1} \frac{N!(N-k)}{k!(N-k)!}$$

$$= MN \sum_{k=0}^{N} \left[\frac{N!}{k!(N-k)!} \right] - \left. \frac{MNN!}{k!(N-k)!} \right|_{k=0,\,N}$$

$$- M \sum_{k=0}^{N} \left[\frac{kN!}{k!(N-k)!} \right] + \left. \frac{MkN!}{k!(N-k)!} \right|_{k=0,\,N} \qquad (F\text{-}1)$$

where the limits on the summation have been changed so that the binomial expansion may be used.

Binomial Expansion

$$(1+x)^N = 1+Nx+ \frac{N(N-1)x^2}{2!} + \frac{N(N-1)(N-2)x^3}{3!} +\ldots = \sum_{k=0}^{N} \frac{N!x^k}{k!(N-k)!} \qquad (F\text{-}2)$$

As $x \rightarrow 1$, the series in equation (F-2) reduces to the first expression on the right of equation (F-1), so equation (F-1) becomes

$$\text{total comparisons} = MN[2^N] - 2MN - M \sum_{k=0}^{N} \frac{kN!}{k!(N-k)!} + MN$$

$$= MN[2^N - 1] - M \sum_{k=0}^{N} \frac{kN!}{k!(N-k)!} \qquad \text{(F-3)}$$

If it were not for the factor k appearing in each term of the last sum, this again would be the binomial series and hence a trivial sum. Let x be any arbitrary parameter. Using standard techniques [56, 57] the additional factor k may be produced by differentiation.

$$kx^k = x \frac{\partial}{\partial x} (x^k) \qquad \text{or}$$

$$\sum_{k=0}^{N} \frac{N!(k)x^k}{k!(N-k)!} = \sum_{k=0}^{N} \frac{N!}{k!(N-k)!} \left[x \frac{\partial}{\partial x} (x^k) \right] \qquad \text{(F-4)}$$

By interchanging the order of summation and differentiation equation (F-4) becomes

$$x \frac{\partial}{\partial x} \sum_{k=0}^{N} \frac{N!\, x^k}{k!(N-k)!} = x \frac{\partial}{\partial x} (1+x)^N$$

$$= xN(1+x)^{N-1} \qquad \text{(F-5)}$$

Again, for the binomial series to approach the series in question, x must approach 1. Taking this limit yields

$$\lim_{x \rightarrow 0} xN(1+x)^{N-1} = N \left[2^{N-1} \right]$$

and, hence, from equation (F-3)

$$\text{total cost comparisons} = NM\left[2^N - 1\right] - NM\left[2^{N-1}\right]$$

$$= NM\left[2^N - 2^{N-1} - 1\right]$$

$$= NM\left[2^{N-1} - 1\right] \qquad \text{(F-6)}$$

BIBLIOGRAPHY

1. Taylor, Robert S. "Change." In: National Information Retrieval Colliquim. 10th, Philadelphia, Pennsylvania, 3-4 May 1973. Proceedings. Changing Patterns in Information Retrieval. Edited by Carol Fenichel. American Society for Information Science, Washington, D. C., 1974. 175 pp.

2. Bar-Hillel, Yehoshua. "Is Information Retrieval Approaching a Crisis?" American Documentation, 14:2 (April 1963) 95-98. (Also in: Yehoshua Bar-Hillel, Language and Information, Addison-Wesley Publishing Company, Reading, Massachusetts, 1964, 365-372).

3. Williams, Martha and Ted Brandhorst. "Data About Data Bases." Bulletin of the American Society for Information Science, 3:2, (December 1976) 20-21.

4. Radwin, Mark. "Nets Make Reference Data Available in 50 Cities." Computerworld, [Boston] 9:48 (November 26, 1975) S/7.

5. "Data Networks." Computerworld [Boston], (November 26, 1975) S/2-S/23. (This is a special section which includes nine reports relating to communications carriers.)

6. Taylor, Robert S. "Question-Negotiation and Information Seeking in Libraries." College and Research Libraries, 29:3, (May 1968) 178-194. (Reprinted in: Key Papers in Information Science. Edited by Arthur Elias. American Society for Information Science, Washington, D. C., 1971, 36-55).

7. Maron, M. E. and J. L. Kuhns. "On Relevance, Probalilistic Indexing and Information Retrieval." Journal of the ACM, 7:3, (July 1960) 215-244.

8. Salton, Gerard. Automatic Information Organization and Retrieval. McGraw-Hill, New York, New York, 1968, 202-234.

9. Kochen, Manfred. Principles of Information Retrieval. Melville Publishing Company, Los Angeles, California, 1974. 203 pp.

10. Salton, Gerard. "Cataloging Software Packages for Automatic Document Processing." In: ASIS Annual Meeting, 39th, San Francisco, California, 4-9 October 1976. Proceedings. Information*Politics. Compiled by Susan K. Martin. American Society for Information Science, Washington, D. C., 1976, Part Two: Full Papers (V-A14) 394-411.

11. Rocchio, J. J. and G. Salton. "Information Search Optimization and Interactive Retrieval Techniques." In: AFIPS Fall Joint Computer Conference. Las Vegas, Nevada, November 1965. Proceedings. Spartan Books, New York, 1965. 293-305.

12. Chang, Y. K., Cirillo, C., and Razon, J. "Evaluation of Feedback Retrieval Using Modified Freezing, Residual Collection, and Test and Control Groups." In: The Smart Retrieval System. Edited by Gerard Salton. Prentice-Hall, Englewood Cliffs, New Jersey, 1971, 355-370.

13. Swanson, Rowena W. "On Terminology for Search Strategies." Journal of the American Society for Information Science, 23:2, (March-April 1972) 133-34.

14. Brandhorst, W. T. "Simulation of Boolean Logic Constraints through the Use of Term Weights." American Documentation, 17:3, (July 1966) 145-6.

15. Angione, P. V. "On the Equivalence of Boolean and Weighted Searching Based on the Convertibility of Query Forms." Journal of the American Society for Information Science, 26:2, (March-April 1975) 112-124.

16. Tenenbein, Aaron and Jay-Louise Weldon. "Probability Distributions and Search Schemes." Information Storage and Retrieval, 10:7-8, (1974) 237-242.

17. Dyke, H. G. "A Figure-of-Merit Ordering System for a Search Output." American Documentation, 10:1, (1959) 85-86.

18. Verhoeff, J., Goffman, W., and Belzer, Jack. "Inefficiency of the Use of Boolean Functions for Information Retrieval Systems." Communications of the ACM, 4, (December 1961) 557-558, 594.

19. Barkla, J. K. "Construction of Weighted Term Profiles by Measuring Frequency and Specificity in Relevant Items." 2nd International Conference on Mechanized Information Storage and Retrieval Systems, Cranfield, 1968.

20. Mulvihill, John and Everett H. Brenner. "Ranking Boolean Search Output." American Documentation, 19, (April 1968) 204-5.

21. Uhlmann, Wolfram, "Document Specification and Search Strategy Using Basic Intersections and the Probability Measure of Sets." American Documentation, 19, (July 1968) 240-6.

22. Miller, William L., "A Probabilistic Search Strategy for Medline." Journal of Documentation, 27:4 (1971) 254-66.

23. Sparck Jones, K. "A Statistical Interpretation of Term Specificity and Its Application in Retrieval." Journal of Documentation, 28, (1972) 11-21.

24. Enslow, P. H., Jr., "A Bibliography of Search Theory and Reconnaissance Theory Literature." Naval Research Logistics Quarterly, 13:2, (June 1966) 177-202.

25. Dobbie, James M., "Search Theory: A Sequential Approach." Naval Research Logisitics Quarterly, 10:4, (December 1963) 323-34.

26. Leimkuhler, Ferdinand F., "A Literature Search and File Organization Model." American Documentation, 19, (April 1968) 131-36.

27. Kraft, Donald H., "A Decision Theory View of the Information Retrieval Situation: An Operations Research Approach." Journal of the American Society for Information Science, 24:5, (September-October 1973) 368-76.

28. See [7].

29. Korfhage, R. R., "A Note on a Relevance Estimate and Its Improvement." Communications of the ACM, 11, (1968) 756.

30. Tague, Jean M., "A Bayesian Approach to Interactive Retrieval." Information Storage and Retrieval, 9, (1973) 129-42.

31. Robertson, S. E. and K. Sparck Jones, "Relevance Weighting of Search Terms." Journal of the American Society for Information Science, 27:3, (May-June 1976) 129-46.

32. Baker, N. R., "Toward an Analytical Framework for Service Organizations." XIV International Meeting of TIMS, Mexico City, (August 1967).

33. Baker, N. R., "Optimal User Search Sequences and Implications for Information System Operation." American Documentation, 20:3, (July 1969) 203-12.

34. Yu, C. T., "A Methodology for the Construction of Term Classes." Information Storage and Retrieval, 10, (1974) 243-51.

35. Cooper, W. S. and M. E. Maron, "Foundations of Probabilistic and Utility-Theoretic Indexing." (Unpublished Draft 4, University of California, Berkeley, June 15, 1976) 33p.

36. Hadley, G., "Sequential Decision Problems." In: Introduction to Probability and Statistical Decision Theory. Holden-Day, San Francisco, 1967, 548-70.

37. Lin S. and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem." Operations Research, 21:2, (March-April 1973) 498-516.

38. Debons, Anthony, ed. Information Science: Search for Identity. Marcel Dekker, New York, 1974. (NATO Advanced Study Institute in Information Science, Proceedings.) 481p.

39. Cooper W. S., "On Selecting a Measure of Retrieval Effectiveness." Journal of the American Society for Information Science, 24:2, (March-April 1973) 87-100.

40. Cooper, William S. "The Inadequacy of Probability of Usefulness as a Ranking Criterion for Retrieval System Output." (Unpublished paper, School of Librarianship, University of California, Berkeley, October 25, 1972) 17p.

41. Marchant, Maurice P. "Strategies for Teaching the Optimizing of Search Strategies." Journal of Education for Librarianship, 16:3, (Winter 1976) 155-61.

42. Klonk, Juergen. "Considerations on How to Obtain and How to Use a Matrix of Usefulness in a Document Retrieval System." (Unpublished term paper, Librarianship 241, School of Librarianship, University of California, Berkeley, December 5, 1972) 10p.

43. The Rand Corporation. A Million Random Digits with 100,000 Normal Deviates. Free Press, Glencoe, Illinois, 1955. 400, 200 pp.

44. Nie, Norman H. and Others. SPSS: Statistical Package for the Social Sciences. 2d ed., New York, McGraw-Hill, 1975. 675p.

45. Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley Publishing Company, Reading, Massachusetts, 1974. 470p.

46. Hillier, Frederick S. and Gerald J. Lieberman. Introduction to Operations Research. Holden-Day, Inc., San Francisco, 1967. 639p.

47. Whitehead, Earl Glen Jr. Combinatorial Algorithms. (Notes by Chester Frankfeldt and Alan E. Kaplan.) Courant Institute of Mathematical Sciences, New York University, New York, 1973. 104p.

48. Karp, Richard M. "Reducibility Among Combinatorial Problems." In: Symposium on the Complexity of Computer Computations. Yorktown Heights, New York, 20-22 March 1972. Proceedings. Complexity of Computer Computations. Edited by Raymond E. Miller and James W. Thatcher. Plenum Press, New York, 1972. 85-103.

49. Cook, Stephen A. "The Complexity of Theorem Proving Procedures." Third ACM Symposium on Theory of Computing, 1970. 151-8.

50. Kolata, Gina Bari. "Analysis of Algorithms: Coping with Hard Problems." Science, 186:4163, (8 November 1974) 520-1.

51. Yu, Clement Tak. Theory of Indexing and Classification. Cornell University, Department of Computer Science, Ithaca, New York, August 1973. 226p. (Ph.D. dissertation, Department of Computer Science, Cornell University, Ithaca, New York. TR 73-181).

52. Knuth, Donald E. The Art of Computer Programming Vol. III Sorting and Searching. Addison-Wesley, Reading, Mass., 1968. 149.

53. Floyd, Robert. "Panel Discussion." In: Symposium on the Complexity of Computer Computations. Yorktown Heights, New York, 20-22 March 1972. Proceedings. Complexity of Computer Computations. Edited by Raymond E. Miller and James W. Thatcher. Plenum Press, New York, 1972. 173-4.

54. Baxendale, Phylis. "Content Analysis, Specification, and Control." In: Annual Review of Information Science and Technology, Vol. 1. Edited by Carlos Cuadra. Wiley, New York, 1966, 73.

55. Herner, Saul and Mary Herner. "Information Needs and Uses." In: Annual Review of Information Science and Technology, Vol. 2. Edited by Carlos Cuadra. Wiley, New York, 1967, 30-1.

56. Mathews, Jon and R. L. Walker. Mathematical Methods of Physics. W. A. Benjamin, Inc., New York, 1964. 49-50.

57. Reif, Frederick. Fundamentals of Statistical and Thermal Physics. McGraw-Hill, New York, 1965. 14-15.